KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

KOMPIUTERIŲ KATEDRA

Audrius Leipus

Arvydas Vapsva

Mobiliųjų E-rinkimų taikomoji sistema

Magistro darbas

Darbo vadovas

doc. V. Kiauleikis

Kaunas

2005

Turinys

1. Įvadas 4
2. Analizės dalis 5
2.1. Tyrimo sritis, objektas, problema, darbo aktualumas 5
2.2. Analizės metodų ir priemonių parinkimas 5
2.3. Esamos situacijos analizė 6
2.4. Literatūros šaltiniuose pateiktų sprendimų problemai spręsti lyginamoji analizė 9
2.5. Projekto tikslas ir jo pagrindimas, kokybės kriterijų apibrėžimas 14
2.5.1. Projektuojamas objektas 14
2.5.2. Projektuojamo objekto paskirtis 14
2.5.3. Projektuojamo objekto funkcijos 14
2.5.4. Reikalavimai projektuojamo objekto posistemėms 15
2.5.5. Eksploataciniai reikalavimai 17
2.5.6. Reikalavimai projekto dokumentacijai 17
2.5.7. Reikalavimai realizacijai 17
2.6. Kompiuterizuojamos sistemos varianto parinkimas 18
2.7. Analizės išvados 23
3. Projekto dalis 24
3.1. Techninė užduotis 24
3.1.1. Techninės dalies projektas 25
3.1.2. Projektuojamo objekto konceptuali schema 26
3.1.2. Projektuojamo objekto konceptuali schema ir aprašymas 26
3.1.3. Darbo vietų sąsajų su specifikuotomis funkcijomis lentelė 30
3.1.4. Kompiuterių darbo vietoms parinkimas ir pagrindimas 30
3.1.5. Aparatinės priemonės 32
3.1.6. Programinės priemonės 32
3.1.7. Patalpų projektas su darbo vietų ir kitos aparatūros išdėstymu 33
3.1.7. Techninės tinklo priemonės 35
3.1.8. Tinklo operacinės sistemos parinkimas, pagrindimas 37
3.2. Reikalavimų modelis 38
3.2.1. Vartotojų panaudojimo atvejų diagrama 38
3.2.2. Dalykinės srities klasių diagrama 39
3.2.3. Vartotojo interfeiso modelis 40
3.3. Sistemos projektas 41
3.3.1. Projekto tikslas 41
3.3.2. Sistemos panaudojimo atvejų diagramos 41
3.3.3. Panaudojimo atvejų scenarijų diagramos 43
3.3.4. Sistemos architektūros modelis 44
3.3.5. Sistemos veiklos modelis 45
3.3.6. Realizacijos modelis 46
3.3.7. Komponentinis sistemos modelis 47
3.3.8. Duomenų bazės modelis 50
3.2. Projekto išvados 51
4. Eksperimentinis tyrimas 52
4.1. Eksperimentinė projekto dalis 52
4.2. Tolimesnio sistemos tobulinimo, plėtojimo galimybės 67
5. Išvados 68

6. Literatūra 69
7. Terminų ir santrumpų žodynas 71
8. Santrauka anglų kalba 73

1. Įvadas

Technologijų progresas toks spartus ir neišvengiamas, kad net sunku išsivaizduoti, ką mokslininkai ir inžinieriai pasiūlys ar pakeis jau rytoj. Visgi yra dalykai, kurie yra nusistovėję ir kurių keitimas ar tobulinimas atrodo ne tik nebūtinas, bet netgi ir neįmanomas. Motyvacija paprasta – pasitikėjimo naujovėmis trūkumas, biurokratizmas ar, galų gale, finansų stoka.

Seimo, Prezidento ar kitos valdžios rinkimų tvarka atrodo jau nusistovėjusi ir tapusi norma. Nepaisant dažnai girdimų įtarimų dėl rezultatų klastojimų ar kitų rinkimų pažeidimų nėra garsiai kalbama apie galimybę iš esmės pakeisti rinkimo procedūras, siekiant užkirsti galimybę piktnaudžiauti. Vis mažėjantis rinkėjų skaičius taipogi kelia nerimą. Trūkumų yra - ir daug... Tai ir paskatino ištirti galimybę realizuoti elektroninių rinkimų modelį remiantis užsienio patirtimi, esamomis technologijomis bei oficialia rinkimus administruojančia institucija. Be abejo, tai nėra tik bandymas adaptuoti esamą rinkimų sistemą – atlikome tyrimą, projektavimą bei analizę, ir galime teigti, kad tai pasiteisina.

2. Analizės dalis

2.1. Tyrimo sritis, objektas, problema, darbo aktualumas

Mobiliųjų E-rinkimų taikomoji sistema - programinis ir techninis sprendimas kompiuterizuotiems, mobiliems rinkimams organizuoti. Objekto paskirtis - balsavimo procedūrų vykdymas, suteikiant vartotojui galimybę keliais būdais atiduoti savo balsą, maksimaliai užtikrinant duomenų saugumą.

Rinkimai suteikia galimybę šalies piliečiams išreikšti savo norus, išsirinkti norimus valdininkus. Tačiau vis dažniau rinkimai tampa korumpuoti ir tendencingi, tai įvyksta, kai konkretus asmuo naudodamas įvairius neleistinus metodus stengiasi nulemti rinkimų eigą savo naudai. Šioms problemoms spręsti yra daug būdų, tačiau vienas geriausių, tai rinkimų sistemos kompiuterizavimas bei mobilizavimas. Panaudojus modernias technologijas galima padaryti rinkimus žymiai skaidresnius, supaprastinti jų eigą, sumažinti jų sąnaudas, pašalinti žmogiškąjį faktorių, kuris dažnai turi įtakos rinkimų eigai ir rezultatams. Tam tikslui projektuojama mobiliųjų E-rinkimų sistema, kuri susideda iš centralizuoto tinklo, kuriame visa informacija surenkama iš rinkėjų tiesiogiai, be tarpininkų. Nemažai investavus į mobiliųjų E-rinkimų sistemą ateityje galima sutaupyti žymiai didesnes pinigų sumas. Naudojantis aukštomis technologijomis galimas maksimalus saugumo užtikrinimas ir ypatingai griežta klaidų kontrolė.

Rinkėjui suteikiama galimybė balsuoti keliais būdais, kad nereiktų stovėti eilėse prie balsavimo urnų. Tai dar labiau skatina atlikti pilietinę pareigą. Mobiliųjų E-rinkimų sistema rinkimams suteiks skaidrumo, aiškumo ir konkretumo. Rinkimų rezultatai iš be tarpininkų pateks į rinkimines apygardas.

2.2. Analizės metodų ir priemonių parinkimas

Projekto procesų analizei ir modeliavimui išsirinkome šiuolaikinę projektavimo kalbą UML naudodamiesi Rational Rose paketu. Ši programa naudojama sudėtingų procesų analizei, dokumentacijai ir klaidų taisymui. Produktas suteikia galimybę aiškiai išdėstyti tokius svarbius faktorius, kaip – kokios veiklos sritys reikalingos, kaip jos įgyvendinamos ir kokie ištekliai reikalingi tai atlikti. Tai suteikia galimybę visą organizacijos modelį atvaizduoti aiškiomis ryšių diagramomis. Šio paketo privalumai :

- Supaprastėja komunikacija, visi kalba ta pačia kalba, iššvaistoma mažiau laiko;
- Reikalavimai lengviau apibrėžiami ir dokumentuojami, mažiau “pamirštų” vietų;
- Vartotojai įtraukiami į programos kūrimą nuo pat pradžių, mažiau perdarymų pabaigoje;
- Priemonė išsaugoti sukauptas žinias įmonėje, net jei žmonės ją palieka;
- Sutaupo laiko susipažįstant su jau sukurtomis sistemomis.

2.3. Esamos situacijos analizė

Rinkimų procesą skirstome į šias pagrindines pakopas :

- Visų Lietuvos gyventojų duomenų bazės sudarymas ir perkėlimas į centrinę darbo stotį;
- Iš visų Lietuvos gyventojų duomenų bazės formuojamas aktyvių rinkėjų (LR piliečių virš 18 metų) sąrašas;
- Rinkėjų sąrašas paskirstomas pagal gyvenamąją vietą konkrečioms rinkiminėms apygardoms;
- Prie centrinės darbo stoties prijungta spausdinimo įranga spausdinami kvietimai į rinkimus pagal konkrečias rinkimines apygardas;
- Kvietimai išdalinami rinkėjams;
- Kiekvienas rinkėjas balsuoja nustatytoje rinkiminėje apylinkėje;
- Atėjus į rinkiminę apylinkę, asmuo identifikuojamas ir jam suteikiama teisė balsuoti;
- Asmuo užbraukia norimą kandidatą lapelyje ir įmeta lapelį į balsadėžę.
- Galimas išankstinis balsavimas paštu;

Įstatyminės rinkimų bazės apžvalga :

Rinkimai vykdomi tam, kad sudaryti valstybės valdymo aparatą. Demokratinės respublikos piliečiai turi prigimtinę konstitucinę teisę dalyvauti valstybės valdyme, ir ši teisė yra nepaneigiama bei neginčijama, išskyrus įstatymo numatytus atvejus. Kaip ir daugelis procesų susijusių su valstybės valdymo veikla, rinkimai yra griežtai aprašyti ir reglamentuoti įstatymų. Kiekvienas žingsnis yra aiškiai apibrėžtas ir detalizuotas, numatytos net pačios smulkiausios detalės. Tai daroma tam, kad būtų aiški rinkimų organizavimo, rengimo ir rezultatų skaičiavimo bei pateikimo visuomenei veiksmų seka. Lietuvos Respublikoje rinkimų būdu į savo postus išrenkamas LR Prezidentas ir LR Seimo nariai. Seimo rinkimus reglamentuoja LR Seimo Rinkimų Įstatymas, o Prezidento rinkimus – LR Prezidento Rinkimų Įstatymas [2].

LR Seimo Rinkimų Įstatymą sudaro vienuolika skirsnių (kurie yra suskirstyti į 98 smulkesnius straipsnius) [1]:

Galima paminėti, kad šis įstatymas užima net 42 puslapius A4 formatu. Naujausia LR Seimo rinkimų Įstatymo redakcija įsigaliojo nuo 2003 05 01.

LR Prezidento Rinkimų Įstatymą sudaro devyni skirsniai (kurie yra suskirstyti į 74 smulkesnius straipsnius) [2].

LR Prezidento rinkimų Įstatymo redakcija įsigaliojo nuo 2002 06 20. Šį įstatymą sudaro 31 puslapis A4 formatu.

Mūsų projektui aktualūs visi įstatymų punktai, kadangi jais remiantis organizuojami rinkimai. Be abejo neišvengiamos įstatymų pataisos norint detaliai reglamentuoti naująją elektroninių rinkimų sistemą, pagrindiniai pakeitimai turėtų būti atlikti šiuose punktuose :

Rinkimų apygardų sudarymas

Atsižvelgiant į patogumą rinkėjui atvykti į balsavimo patalpas ir rinkėjų skaičių, miestų, rajonų teritorijos dalijamos į rinkimų apygardas.

Rinkimų komisijos

Rinkimus organizuoja ir vykdo:

1. Vyriausioji rinkimų komisija;
2. Apygardų rinkimų komisijos.

Rinkėjų sąrašai

1. Rinkimams organizuoti ir vykdyti sudaromi šie rinkėjų sąrašai:
    a. Lietuvos Respublikos rinkėjų sąrašas;
    b. Rinkimų apygardų rinkėjų sąrašai.

Iš Lietuvos Respublikos rinkėjų sąrašo turi būti išbraukiamas miręs LR pilietis, asmuo, netekęs LR pilietybės ir LR pilietis, dėl kurio įsigaliojo teismo sprendimas pripažinti jį neveiksniu.

Rinkėjo pažymėjimas

Rinkėjo pažymėjimas yra rinkimų komisijos išduotas dokumentas, kuriame nurodoma, kurios rinkimų apygardos rinkėjų sąraše yra įrašytas Lietuvos Respublikos pilietis.

Rinkėjo pažymėjime nurodoma:

- rinkėjo vardas ir pavardė;
- rinkėjo gimimo data (metai, mėnuo, diena);
- rinkėjo adresas;
- rinkimų apygardos, į kurios rinkėjų sąrašą įtrauktas rinkėjas, pavadinimas, numeris ir balsavimo patalpos adresas;
- rinkėjo eilės numeris rinkimų apygardos rinkėjų sąraše;
- rinkimų data, balsavimo rinkimų apygardos balsavimo patalpoje laikas.

Jeigu rinkėjas prašo išduoti rinkėjo pažymėjimo dublikatą vietoj pamesto ar negauto rinkėjo pažymėjimo, dublikatas turi būti nedelsiant išduodamas rinkėjui, kai tik yra nustatomi rinkėjo duomenys, kurie turi būti įrašomi į pažymėjimą.

Balsavimo laikas ir vieta

Balsavimas vyksta rinkimų dieną nuo 7 iki 20 valandos apygardos rinkimų komisijos nurodytose terminalų vietose. Rinkėjas balsuoja toje rinkimų apygardoje, į kurios rinkėjų sąrašus jis yra įrašytas, jei kitko nenumato šis įstatymas.

Draudimas rengti balsavimo patalpoje kitus renginius

Balsavimo patalpoje negalima rengti jokių kitų renginių, išskyrus rinkimų organizavimą ir balsavimą. Jokių renginių taip pat negalima rengti ir perėjimo patalpose (koridoriuose) bei prie įėjimo į balsavimo patalpos pastatą.

Rinkėjo asmenybės nustatymas

Prieš įeidamas į mobiliojo balsavimo terminalo patalpą rinkėjas privalo perbraukti rinkėjo pažymėjimą per duryse įmontuotą įrenginį. Taip užtikrinama kad pašaliniai asmenys negalėtų įeiti į balsavimo patalpą. Vėliau jis identifikuojamas naudodamasis balsavimo pažymėjimu bei asmens tapatybės kortele. Jei rinkėjas identifikuotas sėkmingai, jam suteikiama teisė balsuoti.

Galimi pakeitimai

Norint reformuoti rinkimų sistemą į mobiliųjų E-rinkimų sistemą būtinas naujų teisinių aktų priėmimas, bei rinkimų įstatymų pataisos, tam kad būtų galima suformuluoti naują rinkimų vykdymo eigą. Kompiuterizuojant rinkimus keičiasi ne tik jų balsavimo būdas, bet ir rinkimų organizavimo modelis, rinkėjų sąrašų sudarymo metodai, rinkėjų pažymėjimo forma ir paskirtis, apygardos sąvoka, balsų skaičiavimo procesas ir rezultatų pateikimo formos. Todėl visus pasikeitimus reikia detaliai pateikti naujajame pataisytame įstatymų projekte.

Šiai sistemai realizuoti būtina centrinė darbo stotis, kuri talpina visų gyventojų duomenų bazę ir ją apdoroja, bei paskirsto duomenis apygardų darbo stotims. Apygardų darbo stotis apdoroja ir paskirsto apygardos gyventojų duomenų bazę pagal piliečių gyvenamąją vietą. Apygarda – tai ne daugiau 100000 gyventojų apimanti sritis apibrėžta geografiškai. Aparatinė dalis yra trijų lygmenų – centrinės darbo stoties resursai ir visų Lietuvos apygardų darbo stočių tinklas, kuris periodiškai aktyvuojamas informacijai parsisiųsti

į apygardų darbo stotis. Taip užtikrinamas saugumas nuo įsilaužimo į duomenų bazes Internetu, rinkimų eigoje. Apygardos darbo stotis atlieka informacijos pasikeitimo funkciją su centrine darbo stotimi. Trečiasis lygmuo – konkrečiai apygardos darbo stočiai priklausantys balsavimų terminalai, kurie identifikuoja rinkėjus, priima balsus už kandidatus ir saugo sukauptus duomenis per visą rinkimų eigą. Balsavimų terminalai - jų pagalba žmonės gali išrinkti norimą kandidatą prisilietimams jautriame ekrane. Prisilietimams jautrūs ekranai turi atitikti aukščiausios kokybės reikalavimus (ISO standartą Europoje), pagal tai ir bus pasirinktas jų gamintojas. Šie ekranai turės tiesioginę sąsają su duomenų baze, per atitinkamas tvarkykles, kad balsai būtų iškart įtraukiami į ją. Saugumui užtikrinti vykdomas dvigubas balsuojančio asmens identifikavimas. Nuskaitomi duomenys iš asmens identifikavimo kortelės, taip pat nuskaitomas „Bar kodas“ esantis balsavimo biuletenyje. Reikėtų paminėti, kad visi įrenginiai turi būti aprūpinti rimta, saugumą užtikrinančia programine įranga, be to reikia atsižvelgti ir į fizinį jų saugumą, kad pašaliniai asmenys negalėtų prie jų prieiti.

2.4. Literatūros šaltiniuose pateiktų sprendimų problemai spręsti lyginamoji analizė

Jau seniai egzistuoja įvairios rinkimų sistemos, tačiau kiekviena jų turi savų trūkumų.

Analogiškos sistemos

Diebold Rinkimų sistemų sprendimai sukurti JAV pasižymi AccuVote-TS technologija. Ši technologija valdo visą rinkimų procesą, kuris vyksta rinkėjui renkant pretendentą prisilietimams jautriame ekrane. Rinkėjo balsas fiksuojamas virtualioje balsų saugykloje, kuri rinkimams pasibaigus perduodama į darbo stotį ir tada skaičiuojami galutiniai rinkimų rezultatai. Ši sistema yra pritaikyta ir neįgaliesiems, bei neraštingiems žmonėms. Šiame projekte naudojama Global Election Management System (GEMS) programinė įranga, kuri valdo Diebold‘s AccuVote-TS prisilietimams jautrų monitorių. Pritaikyta vartotojui suprantama sąsaja su balso gidu. [Sistemos aprašymas pateikiamas priede Nr. 5]

Sitema WINvote – tai belaidė technologija. Balsuojama prisilietimams jautriame ekrane. Pagrindinis sistemos privalumas – belaidė technologija, kuri leidžia nutolusiam vartotojui bendrauti su pagrindiniais balsavimo sistemos komponentais. Tikimasi, kad ši sistema taps labai populiari dėl itin mažų išlaidų rengiant rinkimus. Sistema apjungia visas elektroninių rinkimų sistemose išplėtotas technologijas. [Sistemos aprašymas pateikiamas priede Nr. 2]

Unilect Patriot – tai daugiakalbė elektroninė balsavimo sistema, kur prisilietimams jautriame ekrane balsuotojas gali pasirinkti norimą kandidatą. Atskiri terminalai yra prijungiami prie pagrindinio kompiuterio, kad duomenys balsavimo metu būtų perduodami tiesiai jam. [Sistemos aprašymas pateikiamas priede Nr. 3]

Sequoia AVC Edge – šioje sistemoje balsuojama prisilietimams jautriame ekrane. Tai taip pat daugiakalbė balsavimo sistema, kuri duomenis įrašinėja į vidinę atmintį. Rinkėjas įkiša specialią kortelę į šią mašiną ir jam yra suteikiama teisė balsuoti. Balsas įrašomas į „balsadėžės“ vidinę atmintį. Pasibaigus rinkimams vidinės atmintys iš „balsadėžių“ yra išimamos ir vežamos i centrines rinkimų stotis, iš kurių duomenys tinklo pagalba papuola i centrinę rinkiminę balsavimo stotį. Centrinėje balsavimo stotyje duomenys prijungiami prie jau esamų duomenų ir gaunamas rinkimų rezultatas. [Sistemos aprašymas pateikiamas priede Nr. 4]

Sequoia AVC Advantage – balsavimo sistema panaši į Sequoia AVC Edge sistemą. Pagrindiniai skirtumai – balsavimo duomenys išsaugomi akumuliatorių palaikomoje RAM atmintyje, kuri rinkimams pasibaigus išimama ir transportuojama i pagrindinę rinkimų stotį. [Sistemos aprašymas pateikiamas priede Nr. 5]

Hart Intercivic eSlate – elektroninė balsavimo sistema kurioje rinkėjas kandidatą pasirenka ne prisilietimams jauriame ekrane, o sukdamas prie ekrano esantį ratuką ir jį paspausdamas pasiekus norimo kandidato vardą. eSlate aparatas kabeliu yra sujungtas su specialiu Judge's Booth kontroleriu. Kontroleris priima iš kompiuterio duomenis ir siunčia jam atgal, jame yra saugomi balsavimo duomenys. Prie šio kontrolerio gali būti prijunta iki dvylikos eSlate kompiuterių. Rinkiminės apylinkės darbuotojai rinkėjams išduoda atsitiktinai sugeneruotų, keturių skaitmenų prisijungimo kodus, balsavimui JBS sistema. Pasibaigus rinkimams, atmintis iš kontrolerio išimama ir pervežama i pagrindinę rinkiminę apylinkę, kurioje prisumuojami prie jau esamų duomenų. [Sistemos aprašymas pateikiamas priede Nr. 6]

ES&S iVotronic – tai balsavimo apylinkės darbuoto aktyvuojama, daugiakalbė sistema. Balsuojama prisilietimam jauriame ekrane. Prasidedant rinkimams balsavimo apylinkės prižiūrėtojas į šią sistemą įkišą specialią kortelę – taip paleisdamas sistemą balsavimui. Balsuotojai pasirenka norimą kalbą. Pasibaigus rinkimams iš sistemos išimama atmintis, kurioje saugomi rinkimų duomenys ir vežama i centrinę rinkimų apygardą. [Sistemos aprašymas pateikiamas priede Nr. 7]

E-Voting sistema Vokietijoje. 2000m. Vokietijoje buvo sukurtas projektas LDS „Internetiniai rinkimai : naujo tūkstantmečio moderni alternatyva“, kuris laimėjo pirmąją vietą 1999/2000m. Elektroninės valdžios tema vykusiose varžybose surengtose CISCO iniciatyva prižiūrint Vokietijos vidaus reikalų ministrei Brigitte Zypries [9]. Varžybose dalyvavo per 50 dalyvių. Išanalizavusi visų pretendentų projektus nepriklausoma mokslininkų iš Potsdamo Universiteto žiuri priėjo išvados vertinti pretendentus pagal tris kriterijus :

- Internetinių technologijų aplikacijų sustiprinimas modernizuojant administravimą;
- administravimo veiksmų kokybės ir efektyvumo didinimas;
- pagerintas vartotojų sąsajos modelis.

Žiuri pabrėžė modernių technologijų ir organizacinių techninių sprendimų, bei saugumo koncepcijas.
LDS ateities planai :

- surengti tikrus tarybos vadovybės rinkimus per Internetą;
- pasiekti aukštą registracijos technologijų lygį, kad rinkėjas galėtų naudotis savo elektroniniu parašu pagal visus Europos teisės aktus;
- kvalifikuotai apmokyti rinkimų administracijos darbuotojus;
- sukurti elektroninių rinkimų sistemos sertifikavimo modelį;
- atlikti griežtą elektroninių rinkimų sistemų eksperimentinį patikrinimą.

Šios sistemos dėka ketinama surengti rinkimus į Vokietijos parlamentą 2006 m. Ir Europos parlamento rinkimus 2006m. Arba Vokietijos parlamento (Bundestago) elektroninius rinkimus 2006m. Šiuo metu vyksta intensyvūs parengiamieji darbai , kad paruošti elektroninių rinkimų sistemą pagal visus Vokietijos konstitucinius įstatymus bei pasiūlyti išsamią informacinę struktūrą sistemos viduje.

2000m. Surengti rinkimai vien tik Internetu buvo sėkmingi.

2000m birželio mėnesį buvo sėkmingai surengti pirmieji kompanijos tarybos valdžios rinkimai Internetu panaudojant elektroninį parašą. Buvo naudojami tik paprasti kompiuteriai su pelyte bei „smart-card“ tipo kortelėmis, kad užfiksuoti rinkėjo balsą. Taip pat buvo galima balsuoti ir specialiu kompiuteriu viešose balsavimo kabinose, iš kurių biuleteniai buvo Internetu persiunčiami į virtualiąją biuletenių dėžę. Tačiau suskaičiavus rezultatus nustatyta , kad 60% rinkėjų balsavo Internetu naudodamiesi šio projekto programine įranga I-Vote. Balsavusieji naująja technologija liko patenkinti ypatingai patogiu balsavimo būdu.

Šis bandymas įrodė, kad įmanoma surengti elektroninius rinkimus Internetu, užtikrinant patikimą rinkėjų identifikavimą, apsisaugant nuo rezultatų iškreipimo, bei užtikrinant saugumą.
Šio projekto esminiai momentai :

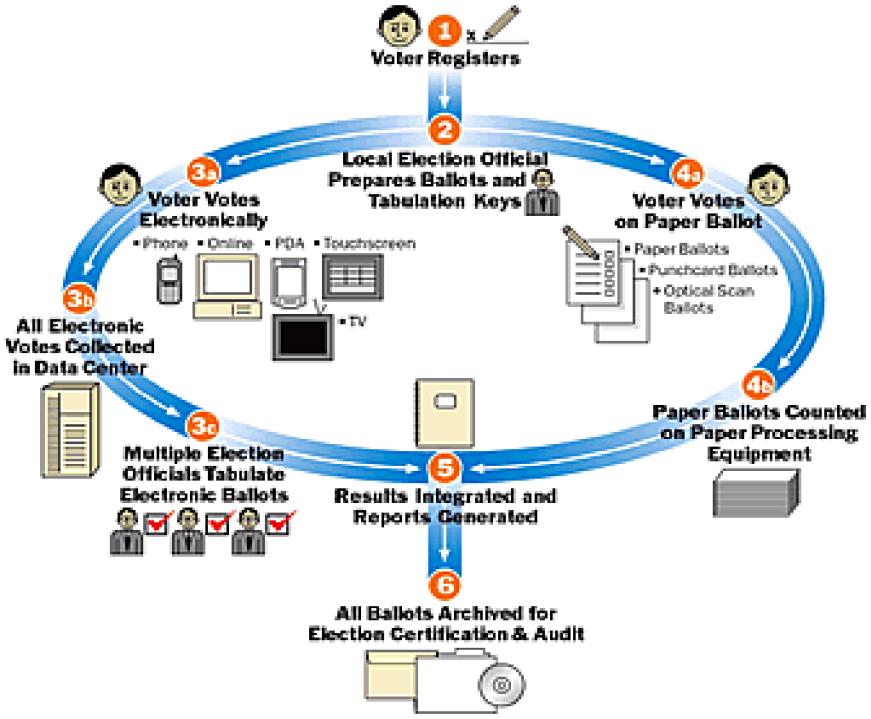
- panaudotas RSA kodavimo algoritmas duomenų apsaugai;
- fizinės ir virtualios balsavimo biuletenių saugyklos imitavimas;
- rinkėjo registracija.

Šis projektas yra bandymų stadijoje, tačiau vyksta sėkmingai.

VoteHere kompanijos E-rinkimų sistemos. VoteHere kompanija yra viena iš pradininkių elektroninių rinkimų sistemų kūrimo srityje ir realiai iš to gaunanti pajamas [6]. Ši JAV bendrovė yra sukūrusi ne vieną E-rinkimų sistemą, pritaikytą tiek įvairių kompanijų vadovybėms, tiek universitetų valdžiai rinkti (1 pav.). Pagrindinė kliūtis su kuria VoteHere susiduria kuriant E-rinkimų sistemas – nesaugus Internetas. Kad

užtikrinti balsavimų privatumą ir saugumą, ši kompanija išskyrė kelis punktus, kuriuos būtina įgyvendinti norint tai pasiekti :

- tinkamumas – tik turintys teisę rinkėjai gali balsuoti;
- unikalumas – galima balsuoti tik vieną kartą;
- saugumas – niekas negali pridėti, pakeisti ar ištrinti balsų nepastebėtas;
- kontrolė – bet kas gali patikrinti ar visi balsai buvo suskaičiuoti teisingai;
- patogumas – balsavimą fiksuoti greitai ir nereikalaujant papildomų įgūdžių;
- paslankumas – įvairūs biuletenių tipai;
- mobilumas;
- efektyvumas – rinkimai rengiami ir prižiūrimi minimaliomis sąnaudomis.

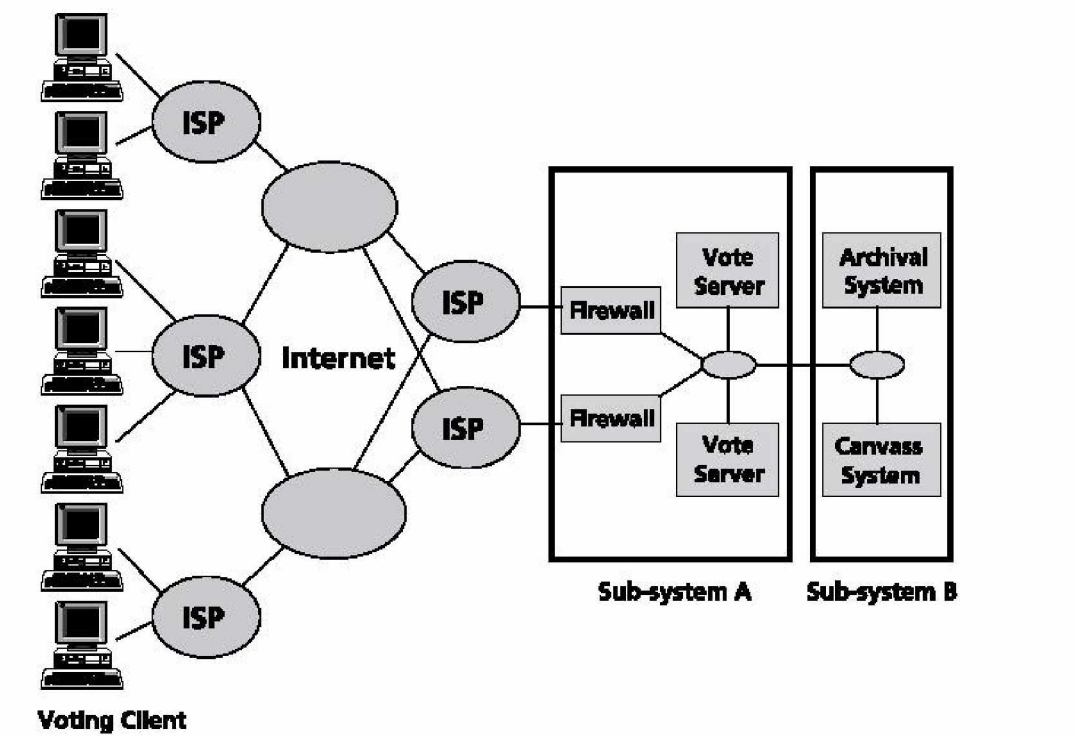


1 pav. VoteHere E-rinkimų ir tradicinių rinkimų schema

„VoteHere“ technologija sukurta naudojant atvirojo kodo architektūrą, kas leidžia ją integruoti daugelyje skirtingo tipo įrenginių su skirtingomis operacinėmis sistemomis. Suteikiama galimybė naudoti bet kokią techninę įrangą ir standartines Interneto naršykles tikriems rinkimams organizuoti. „VoteHere“ technologija iš esmės modernizavo rinkimų procesą. Pagrindiniai sistemos privalumai – supaprastinta rinkimų rezultatų valdymo ir skaičiavimo logistika. Rinkėjui suteikiama galimybė balsuoti iš jam patogios vietos. Rinkimų išlaidų minimizavimas siūlant efektyvų ir trumpą apmokymą administratoriams bei nebrangų įrangos palaikymą ir priežiūrą [5].

IPI – Internet Policy Institute internetinių rinkimų projektas. Nuotoliniai Internetiniai rinkimai patraukė žmonių ir spaudos dėmesį ir yra gretinami su Internetiniais rinkimais balsavimo apygardose, tačiau norint juos įgyvendinti reikia išspręsti daug techninių ir organizavimo elgsenos problemų [6]. Tačiau yra viena tarpinė grandis tarp nuotolinių ir tradicinių rinkimų rūšių – tai rinkimų kioskai, atsparūs sukčiavimams, papirkinėjimams ir klastotėms, kuriuos būtų galima įrengti mokyklose, pašto skyriuose ir netgi prekybos centruose (2 pav.). Vienas šių kioskų privalumas - jie būtų prižiūrimi rinkimų komisijos darbuotojų. Balsavimas kioskuose būtų stebimas rinkimų komisijos darbuotojų, tyrinėtojų ar netgi filmavimo kamerų, kad užtikrinti saugumą ir slaptumą bei užkirsti kelią prievartiniam balsavimui ir kitoms įsikišimo formoms.

Balsavimo kioskuose yra daug saugumo problemų, tačiau labiau priimtinos nei problemos susijusios su nuotoliniais rinkimais.



2 pav. Bendra internetinių rinkimų schema

Valdžios struktūrų rinkimuose balsavimo klientai bus apygardų balsavimo terminalai, tuo tarpu nuotolinėje rinkimų sistemoje balsavimo klientai bus – individualūs kompiuteriai namuose ar darbo vietoje. Šie balsavimo klientai pajungti prie vieno ar kelių Interneto paslaugų tiekėjų teikiančių Internetą ir darbo stočių prieigas Darbo stoties prieigos yra dalijamos į dvi dalis : posistemė A, kuri surenka užkoduotus biuletenius ir posistemė B, kuri atkoduoja biuletenius, juos skaičiuoja ir archyvuoja bei sukuria išsamias ataskaitas.

Pateikiama nagrinėtų rinkimų sistemų palyginimo lentelė: *Lentelė Nr. 1*

Balsavimo būdai	VoteHere	DieBold	E-Voting	Unilect Patriot	ES&S iVotronic
Įprastu paštu	Taip	Taip	Ne	Ne	Ne
Telefonu	Taip	Taip	Ne	Ne	Ne
SMS	Taip	Ne	Ne	Ne	Ne
E-mail‘u	Taip	Ne	Taip	Ne	Ne
Terminalu	Taip	Taip	Taip	Taip	Taip
www svetainėje	Taip	Ne	Taip	Ne	Ne
Skaitmeniniu TV	Taip	Ne	Ne	Ne	Ne

Lentelė 2.1. Esamų rinkimų sistemų funkcijų palyginimas

2.5. Projekto tikslas ir jo pagrindimas, kokybės kriterijų apibrėžimas

2.5.1. Projektuojamas objektas

Programinis ir techninis sprendimai kompiuterizuotiems E-rinkimams organizuoti.

2.5.2. Projektuojamo objekto paskirtis

Balsavimo procedūrų vykdymas, maksimaliai užtikrinant duomenų saugumą, suteikiant vartotojams alternatyvaus balsavimo galimybes, bei operatyviai pateikiant informaciją apie rinkimų eigą.

2.5.3. Projektuojamo objekto funkcijos

- Kvietimų į rinkimus spausdinimas;
- Rinkėjo identifikavimas;
- Rinkėjo balso priėmimas;
- Rinkėjo balso fiksavimas duomenų bazėje;
- Rinkėjų balsų sisteminimas;
- Rinkėjų balsų suskaičiavimas;
- Rinkimų rezultatų pateikimas;
- Rinkėjų aktyvumo statistika.

2.5.4. Reikalavimai projektuojamo objekto posistemėms

Reikalavimai aparatūros posistemei:

Centrinė darbo stotis:

1. Reikalingas galingas kompiuteris, prijungtas prie greito Interneto kanalo, atlikti šioms operacijoms:
    - piliečių duomenų bazės apdorojimas;
    - rinkėjų pažymėjimų generavimas;
    - rinkiminių apygardų darbo stočių užklausų apdorojimas;
    - rinkimų eigos stebėjimas ir kontrolė;
    - rinkimų rezultatų apdorojimas, ataskaitų generavimas.
2. Rinkėjų pažymėjimams spausdinti specializuota spausdinimo aparatūra.

Rinkiminės apygardos darbo stotis:

1. Kompiuteris, kuris atliktų šias operacijas:
    - rinkiminei apygardai siunčiamų duomenų apdorojimas;
    - terminalo užklausų apdorojimas;
    - duomenų terminalui siuntimas;
    - periodinis rinkimų statistikos priėmimas, apdorojimas ir siuntimas;
    - rinkimų rezultatų priėmimas, apdorojimas ir siuntimas.

Balsavimo terminalas:

1. Kompiuteris, kuris atliktų šias operacijas:
    - balsuojančių žmonių duomenų apdorojimas;
    - kandidatų duomenų priėmimas;
    - rinkėjo identifikavimas;
    - balso už kandidatą fiksavimas;
    - kandidatų balsų sumavimas;
    - rinkimų statistikos siuntimas apygardos darbo stočiai;
    - rinkimų rezultatų siuntimas apygardos darbo stočiai.
2. Brūkšninio kodo skaitytuvas.

Reikalavimai informacijos posistemei:

Centrinė darbo stotis:

- centrinėje darbo stotyje iš Lietuvos gyventojų duomenų bazės atrenkami piliečiai pagal Lietuvos LR įstatymus galintys dalyvauti rinkimuose;

- gyventojų sąrašas paskirstomas pagal apskritis;
- balsavimo duomenų (statistika, rezultatai) centrinėje darbo stotyje priėmimas, ataskaitų formavimas;
- rinkimų eigos stebėjimas ir kontrolė.

Rinkiminės apygardos darbo stotis:

- gauna tai rinkiminei apygardai priklausantį rinkėjų sąrašą;
- siunčia terminalams priklausančius rinkėjų sąrašus.

Balsavimo terminalas:

- iš rinkiminės apygardos darbo stoties gauna kandidatų sąrašą;
- balsuojančio piliečio atpažinimas ir prisijungimas prie terminalo;
- iš rinkiminės apygardos gauna informaciją apie rinkėją (gali, negali balsuoti ir kodėl, kur kreiptis jei asmens nėra duomenų bazėje);
- balsavusio piliečio pasirinkimo registravimas;
- balsavusio piliečio atpažinimas;
- rinkimų statistikos siuntimas apygardos darbo stočiai;
- rinkimų rezultatų siuntimas apygardos darbo stočiai.

Reikalavimai vartotojo sąsajai:

Centrinė darbo stotis:

1. Aiškumas;
2. Patogumas;
3. Našumas;
4. Intuityvumas.

Apygardos darbo stotis:

1. Aiškumas;
2. Patogumas;
3. Našumas;
4. Intuityvumas.

Balsavimo terminalas:

1. Aiškumas;
2. Patogumas;
3. Našumas;
4. Intuityvumas;
5. Pritaikymas neįgaliesiems žmonėms;
6. Ergonomika.

2.5.5. Eksploataciniai reikalavimai

Rinkėjų informacija imama iš gyventojų registro. Tarpiniai duomenys ir galutiniai rezultatai saugomi taip, kad prie jų galėtų prieiti tik tą teisę turintys asmenys.

2.5.6. Reikalavimai projekto dokumentacijai

Visa rinkimų procedūra turi būti suskirstyta į atskirus etapus (rinkėjo sąrašų generavimas, rinkėjų skirstymas pagal rinkimines apygardas, kvietimų spausdinimas ir t.t.). Kiekvienas jų turi būti pilnai dokumentuotas, t.y. numatant galimus darbų sutrikimus ir reglamentuoti jų sprendimą tiek techninėmis, tiek programinėmis priemonėmis. Instrukcija rinkėjams turi būti kuo trumpesnė ir lengvai suprantama tiek susidūrusiems su kompiuterine technika, tiek ir kompiuterio visiškai neišmanantiems piliečiams. Prieš rinkimus rinkėjai žiniasklaidos dėka (televizija, spauda) turi būti supažindinami su nauja rinkimų vykdymo technologija.

2.5.7. Reikalavimai realizacijai

Realiai pritaikant sistemą reikia įstatymiškai apibrėžti elektroninę-kompiuterinę rinkimų procedūrą. Tiek rinkimų, tiek bandymų metu naudoti tik sertifikuotą ir licenzijuotą kompiuterinę įrangą. Duomenų mainams naudoti techniškai saugius duomenų kanalus, perduodamus duomenis koduoti šiuolaikiškais kodavimo algoritmais.

Programinė įranga turi būti ištestuota prieš naudojant ją tikruose rinkimuose.

2.6. Kompiuterizuojamos sistemos varianto parinkimas

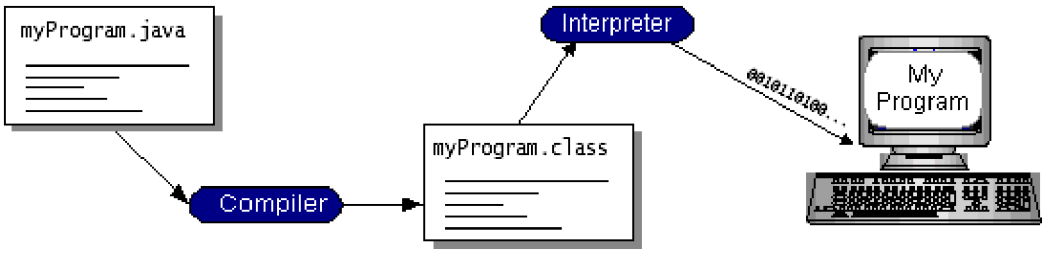
Projektas realizuojamas JAVA programavimo kalba, darant sąsają su PostgreSQL duomenų baze.

Java™ programavimo kalbos apžvalga:

Vis dažniau šiuolaikinėse technologijose galime išgirsti minint Java™ programavimo kalbą. Kuo ji tokia ypatinga, kad taip greitai išpopuliarėjo programuotojų tarpe? Taigi, išsamus Java™ kalbos ypatybių aprašymas. Visų pirma Java™ yra ne tik programavimo kalba, bet ir platforma. Todėl pradžioje apžvelkime Java™ kaip programavimo kalbą – tai aukšto lygio programavimo kalba, kurią būtų galima apibūdinti šiais žodžiais [3]:

- paprasta;
- dinamiška;
- saugi;
- nepriklausoma nuo kompiuterio platformos;
- orientuota į objektinį programavimą;
- tinkama „real-time“ taikymams;
- dinamiška;
- efektyvi (naši).

Ir tai tik maža dalis gerų savybių, kuriomis pasižymi ši programavimo kalba. Daugumoje programavimo kalbų norint, kad programa veiktų jūsų kompiuteryje reikia ją arba sukompiliuoti (sudaryti iš kodo), arba vykdyti (interpretuoti iš kodo). Tuo tarpu Java™ programavimo kalboje programa yra ir kompiliuojama, ir interpretuojama. Pirmiausiai kompiliatoriaus dėka, programa paverčiama į tarpinę kalbą, vadinamą Java™ kodu (Java™ bytecodes) – šis kodas nepriklauso nuo kompiuterio platformos ir yra nuskaitomas ir vykdomas Java™ platformos interpretatoriaus. Interpretatorius nagrinėja ir paleidžia kiekvieną Java™ kodo komandą kompiuteryje. Programos kompiliavimas atliekamas tik kartą, o interpretuojama kaskart kai ji iškviečiama. Šis paveikslėlis labai vaizdžiai iliustruoja šį procesą (3 pav.):



3 pav. Java™ technologijos veikimo principas

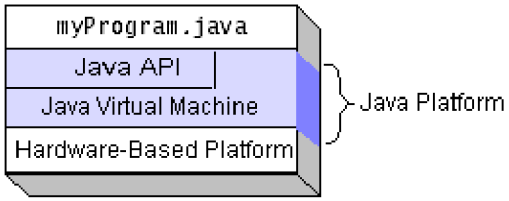
Galima įsivaizduoti Java™ kodą kaip mašininio kodo Java™ Virtual Machine (Java™ VM) instrukcijas. Kiekviena atnaujinta naršyklė turi Java™ VM realizaciją. Java™ kodas padeda išspręsti skirtingų platformų nesuderinamumo problemą. Atsiranda galimybė sukurtą programą sukompiliuoti vienoje platformoje ir paskui ją interpretuoti skirtingos platformos interpretatoriumi, jei jis turi Java™ VM.

Dabar apžvelgsime Java™ platformą. Platforma - tai programinė arba aparatinė aplinka, kurioje realizuojamos programos. Tokios platformos kaip Win2000 arba Linux gali būti apibūdintos kaip operacinės sistemos ir aparatinės dalies derinys. Tuo tarpu Java™ platforma yra grynai programinė platforma, kuri dirba ant aparatinės platformos pagrindo. Java™ platforma susideda iš dviejų dalių :

- Java™ Virtual Machine;
- Java™ Application Programming Interface (API).

Virtuali Java™ mašina sudaro Java™ platformos pagrindą ir ji dirba vienu lygmeniu aukščiau įvairių technine įranga pagrįstų platformų. Tuo tarpu Java™ API – tai platus įvairių reikalingų programinių priemonių rinkinys, kuris yra suskirstytas į giminingų klasių bibliotekas, kurios dar yra žinomos kaip paketai (packages).

Java™ platforma galima pavaizduoti taip (4 pav.) :



4 pav. Java™ platforma

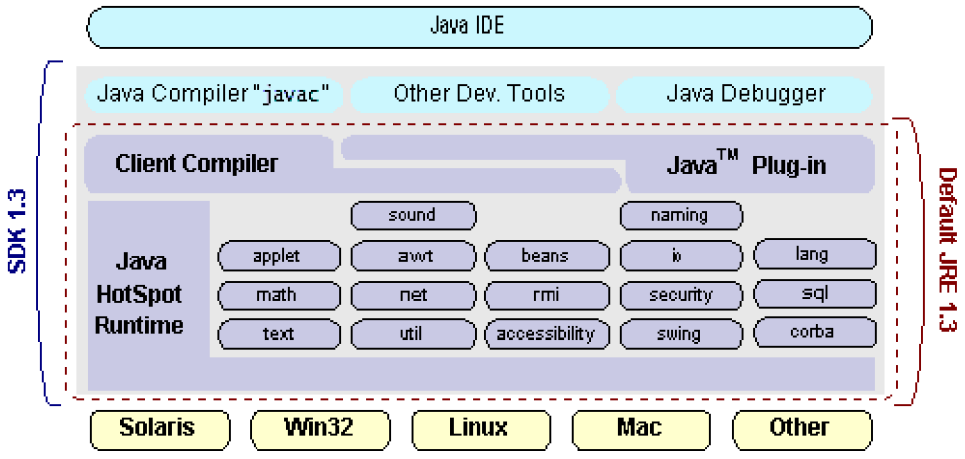
Panagrinėjus Java™ programavimo kalbą ir pačią platformą kyla klausimas, kokius realius projektus galima įgyvendinti naudojant Java™ programavimo kalbą. Pagrindiniai programų tipai parašyti Java™ kalboje tai apletai ir aplikacijos, tačiau jos daugiau skirtos internetiniam naudojimui. Be elementarių programų, naudojant gausias API bibliotekas galima realizuoti ir ypač sudėtingus projektus. Galima kurti

tokias aplikacijas, kaip darbo stotys, kurios aptarnauja ir palaiko klientus internete, servlets – tai aplikacijos veikiančios darbo stoties pusėje. Galima kurti proxy, mail stotis. Taikymų ratas yra ypatingai platus dėl Java™ technologijos paslankumo ir suderinamumo.

Kaip API palaiko toki platų taikymų ratą? Programinių komponentų paketai užtikriną platų funkcionalumą. Kiekviena pilna Java™ platforma suteikia galimybes naudotis šiomis savybėmis :

- pagrindiniai elementai : objektai, eilutės, gijos, skaičiai, įvestis ir išvestis, duomenų struktūros, data ir laikas, ir t.t.;
- apletai : apletų naudojimo privalumai;
- tinklų panaudojimas: URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, ir IP (Internet Protocol) adresai;
- internacionalizacija : pagalba rašant programas, kurios gali būti pritaikomos vartotojams visame pasaulyje. Programos gali automatiškai nusistatyti paleidimo vietą ir būti rodomos tam tikra kalba;
- saugumas : ir žemo ir aukšto lygio, įskaitant elektroninį parašą, viešojo ir privataus rakto valdymas, priėjimo kontrolė ir sertifikatai;
- programiniai komponentai : žinomi kaip JavaBeans™, gali būti įdiegti į egzistuojančią struktūrą;
- Java™ Database Connectivity (JDBCTM): užtikrina vientisą priėjimą prie plataus rato giminingų duomenų bazių.

Taip pat Java™ platforma turi API įgyvendinti 2D ir 3D grafiką, telefoniją, balso atpažinimą, animaciją ir dar daug specifinių priemonių. Sekantis paveikslėlis pavaizduoja kas sudaro Java™ 2 SDK (5 pav.):



5 pav. Java™ SDK schema

Taigi galime daryti išvadas, kad Java™ tai ateinanti naujos kartos programavimo kalba. Todėl pateiksime keletą priežasčių, kodėl mes būtent šią programavimo kalbą pasirinkome savo projekto realizavimui :

- Nepaisant to, kad Java™ yra galinga objektinio programavimo kalba, ją nesudėtinga įsisavinti, ypač tiems, kas yra susidūręs su C++ programavimo kalba;
- Įvairių programavimo kalbų kodo palyginimai rodo, kad Java™ kodas yra keturis kart trumpesnis, nei tos pačios programos kodas C++ kalboje;
- Galima dvigubai sutrumpinti programavimo laiką, kadangi kodas yra trumpesnis ir pati kalba lengviau įsisavinama nei C++;
- Galima išlaikyti savo programas veikiančias visose platformose programuojant tik Java™ kalba;
- Ši programavimo kalba atitinka šiandienos reikalavimus.

Vartotojo sąsaja galutiniame projekto variante taip pat realizuojama Java™ programavimo kalba. Dėl gerų savybių ir suderinamumo su įvairiomis platformomis, mes pasirinkome būtent Java™ programavimo kalbą, tuo labiau, kad ji yra kompleksiška, stabiliai integruojama į kitas sistemas. Pagrindiniai reikalavimai vartotojo sąsajai, tai ypač lengvas valdymas ir aiškumas. Kadangi didelė Lietuvos piliečių dalis yra nemokantys dirbti kompiuteriu, tai būtina vartotojo sąsają padaryti tokią, kad žmogus nesunkiai galėtų suprasti balsavimo procedūrą. Mygtukai turi būti dideli, su aiškiais užrašais, vaizdingi, kad nesumaišyti jų tarpusavyje. Reikia atkreipti dėmesį ir pritaikyti šią sistemą ir neįgaliesiems žmonėms.

PostgreSQL apžvalga :

Duomenų bazė šiame projekte realizuojama PostgreSQL reliaciniais objektais pagrįsta duomenų bazių valdymo sistema (ORDBMS), kuri yra pagrįsta POSTGRES Version 4.2 sistema sukurta JAV Kalifornijos universiteto mokslininkų. PostgreSQL yra atvirojo kodo sistema ir palaiko SQL92/SQL99 programavimo kalbas bei turi daug modernių funkcijų [4]. Bet pagrindinis aspektas dėl kurio mes pasirinkome būtent šią sistemą - tai, kad ji yra nemokama. POSTGRES pirmoji pradėjo taikyti daugelį reliacinių objektų koncepcijų, kurios po truputį atsiranda ir komercinėse duomenų bazių sistemose. Tradicinės duomenų bazių valdymo sistemos (RDBMS) palaiko duomenų modeliavimą susidedantį iš vardinių ryšių, turinčių konkretaus tipo atributus. Šiuolaikinėse komercinėse sistemose galimi tokie duomenų tipai, kaip – slankaus kablelio arba sveikieji skaičiai, ženklų eilutės, pinigai ir datos. Dažnai pripažįstama, kad toks komercinių duomenų bazių modelis yra nepakankamas ateities duomenų apdorojimo programoms. Reliacinis modelis sėkmingai pakeitė ankstesnius modelius dėl savo paprastumo. Tačiau kartais dėl šio paprastumo be galo sunku įgyvendinti

konkrečias aplikacijas. PostgreSQL siūlo esminius papildomus priedus, kurie padeda vartotojui lengvai išplėsti sistemą, tai :

- paveldėjimas;
- duomenų tipai;
- funkcijos.

Šie priedai suteikia sistemai lankstumo ir pajėgumo :

- apribojimai, ryšiai;
- trigeriai;
- taisyklės;
- operacijų integracija.

Visos šios ypatybės pastato PostgreSQL sistemą į reliacinių objektų duomenų bazių kategoriją, nors ji ir turi kai kurių orientuotų objektų duomenų bazių bruožų. Todėl PostgreSQL labai gerai palaiko tradicines reliacines duomenų bazių kūrimo kalbas. Be to ši sistema turi sąsajas su C, C++, Java™, Perl, Tcl ir Python programavimo kalbomis.

PostgreSQL naudoja kliento/darbo stoties modelį. PostgreSQL sesija susideda iš šių sąveikaujančių procesų :

- Darbo stoties procesas, kuris valdo duomenų bazės laikmenas, priima prisijungimo užklausas iš kliento aplikacijų ir vykdo duomenų bazėje veiksmus kliento vardu. Duomenų bazės darbo stoties programa vadinama „postmaster“.
- Vartotojo kliento aplikacija, kuri atlikinėja duomenų bazės operacijas. Kliento aplikacijos gali būti įvairios : klientas gali būti kaip tekstu orientuotas įrankis, grafinė aplikacija, tinklo (web) darbo stotis, kuri jungiasi prie duomenų bazės, kad parodytų tinklo puslapius ar netgi specializuotas duomenų bazių priežiūros įrankis. Kai kurios kliento aplikacijos pateikiamos kartu su PostgreSQL programa, bet daugelį jų susikuria pats vartotojas.

Paprastai kliento/darbo stoties aplikacijose, klientas ir darbo stotis gali būti skirtinguose kompiuteriuose (hosts). Tuo atveju jie tarpusavyje informacija keičiasi TCP/IP protokolo pagalba. Į tai reikia atsižvelgti, nes laikmenos, kurios prieinamos kliento kompiuteryje, gali būti neprieinamos duomenų bazės darbo stotyje. PostgreSQL darbo stotis vienu metu palaiko keleto klientų prisijungimą. Šiam tikslui ji paleidžia naują procesą kiekvienam prisijungimui. Todėl klientas ir naujas procesas tarpusavyje „bendrauja“ be „postmaster“ proceso įsikišimo. Bet „postmaster“ procesas visada lieka aktyvus ir laukia naujo kliento prisijungimo.

Apžvelgus PostgreSQL architektūros pagrindus matome, kad ši duomenų bazių valdymo programa tinka mūsų projektui, dėl savo struktūros ir naujų idėjų, taip pat ji yra nemokama. [Išsamesnis PostgreSQL aprašymas pridedamas priede Nr. 8]

2.7. Analizės išvados

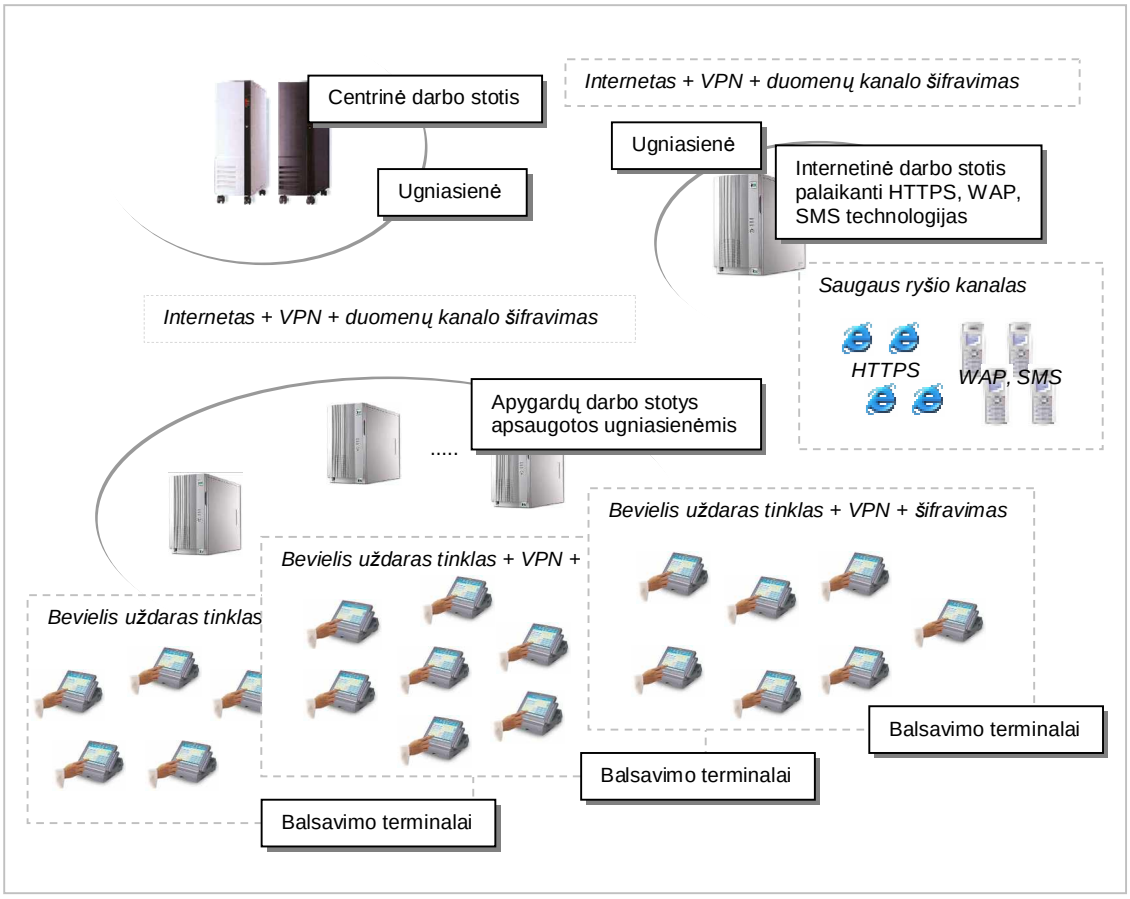
Nėra vieningos balsavimo sistemos kuri apjungtų kelis balsavimo būdus. Esamos rinkiminės sistemos yra pakankamai aukšto lygio, tačiau kiekviena jų apima tik siaurą sritį, nesudarydamos vieningo sprendimo elektroniniams rinkimams organizuoti. Yra įmanoma sukurti tokią sistemą kuri būtu paprasta, daugiafunkcinė, maksimaliai saugi, apjungtų jau esamų idėjų ratą ir taip pat suteiktų galimybę balsuoti naujais būdais. To ir yra siekiama šiame darbe.

3. Projekto dalis

Projekto valdymas atsakinga ir sudėtinga užduotis. Tam, kad išvengti daugybės klaidų realizavimo metu būtina atlikti detalų sistemos projektavimą. Projektavimo metu sudėtingi sistemos mazgai detalizuojami skaidant į paprastesnius segmentus. Techninės ar programinės dalies analizės eiliškumas bendru atveju neturi reikšmės, tačiau šiuo atveju reikia nepamiršti daugelio techninių apribojimų kurie įtakoja programinę realizaciją.

3.1. Techninė užduotis

Reikia apibrėžti kokie pagrindiniai techniniai-programiniai komponentai sudarys projektuojamą sistemą, todėl reikalingas grafinis sistemos planas.



6 pav. Principinė techninio realizavimo schema

Paveikslėlyje matome, jog sistemą sudaro kelios didelės tinklinės sistemos techninėmis bei programinėmis priemonėmis atskirtos viena nuo kitos. Ryšys tarpusavyje palaikomas naudojantis saugiais kanalais. Sekančiuose skyriuose apžvelgsime sistemos dalių ypatybes.

3.1.1. Techninės dalies projektas

Centrinė darbo stotis:

Centrinė darbo stotis skirta duomenų apdorojimui, rinkėjų sąrašų formavimui, duomenų pateikimui, rinkimų eigos stebėjimui bei rezultatų skaičiavimui. Likus kelioms valandoms iki rinkimų *centrinė darbo stotis* atidaro saugų priėjimą prie saugomų duomenų ir atitinkamos darbo stotys prisijungia bei užklausų metu parsisiunčia reikiamus duomenis. Ryšio seansų metu užtikrinamas saugus ryšio kanalas. Rinkimų metu *centrinė darbo stotis* priima statistinius duomenis apie rinkėjų aktyvumą bei stebi tinklo būseną, taip įspėdama apie galimus vienokius ar kitokius trikdžius. Pasibaigus nustatytam laikui rinkimų duomenys priimami į *centrinę darbo stotį* ir gavus visus duomenis bei juos apdorojus pateikiami rezultatai.

Apygardos darbo stotis:

Siekiant sumažinti tinklo apkrovimą pasitelkiamos *apygardos darbo stotys*, kurios veikia kaip tarpininkai tarp centrinės darbo stoties ir balsavimo terminalų. *Apygardos darbo stotis*, gavusi atitinkamą duomenų paketą, juos apdoroja (suskaido į dalis atitinkamiems terminalams) ir toliau persiunčia. Vėliau renka statistikos bei būklės duomenis. Pasibaigus rinkimų laikui tarpininkauja surenkant duomenis iš balsavimo terminalų ir juos pateikia centrinei darbo stočiai.

Toks sprendimas padės papildomai užtikrinti saugumą visiškai atskiriant centrinę darbo stotį nuo balsavimo terminalų tokiu būdu išvengiant nenumatyto duomenų pakeitimo ar kitų trikdžių.

Yra nemaža tikimybė, kad kai kurie rinkėjai rinkimų dieną dėl vienų ar kitų priežasčių neturės balsavimo pažymėjimų, todėl numatoma galimybė, jog *apygardos darbo stotys* įdiegus papildomą programinę įrangą galės tarpininkauti suteikiant balsavimui reikalingą dokumentą. Esant reikalui, numatoma, jog *apygardos darbo stotys* kreipsis į centrinę darbo stotį papildomų duomenų rinkėjo pažymėjimui pagaminti.

Balsavimo terminalas:

Balsavimo terminalas – tai ypatingai daug sąnaudų reikalaujanti sistema. Ji turi ne tik užtikrinti patogią balsavimo sąsają rinkėjui, bet ir būti izoliuota nuo galimų išorės veiksnių. Tai užtikrins bevielio ryšio techninė įranga turinti galimybę apjungti stotis į saugų ir koduojamą tinklą.

Terminalo užduotis – identifikuoti rinkėją bei suteikti jam galimybę saugiai balsuoti. Duomenys privalo būti saugiai talpinami *terminalo* techninėmis priemonėmis. Tai gali būti ne tik elektroninės atminties laikmenos, bet ir papildomi įrenginiai užtikrinantys duomenų saugumą (naudojami papildomam rezultatų tikrumo įrodymui).

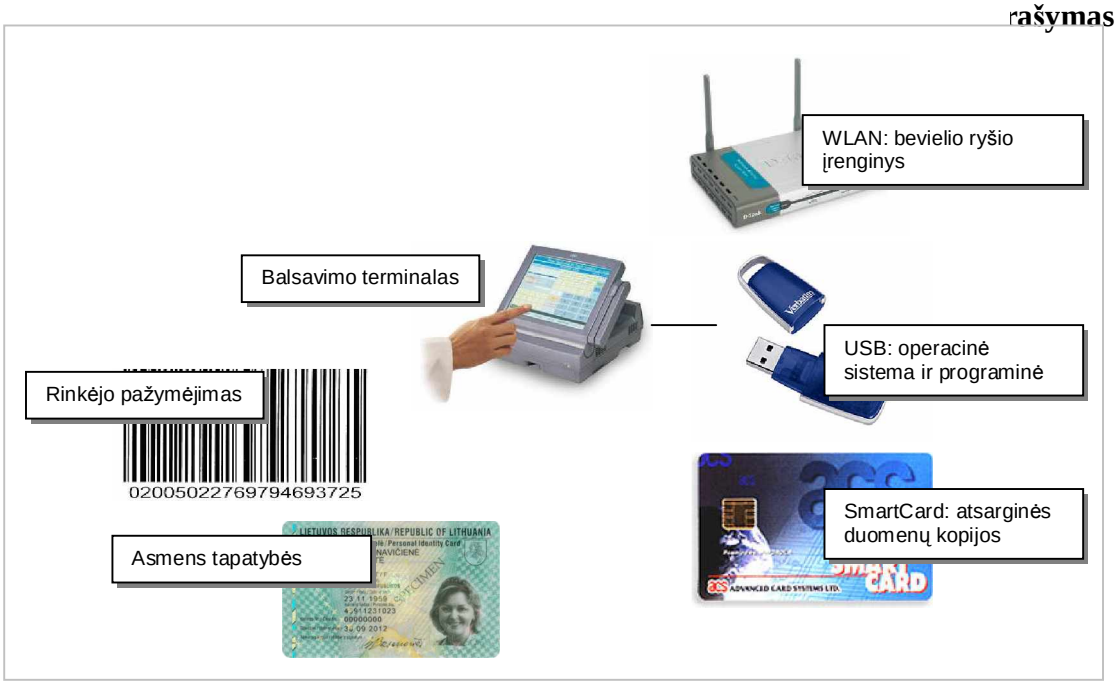
Startavus *terminalą* pastarasis prisijungia prie apygardos darbo stoties ir parsisiunčia reikalingus duomenis rinkėjams identifikuoti, bei duomenis apie kandidatus. Rinkėjas turi galimybę susipažinti su balsavimo procedūros pavyzdžiu paspaudės mygtuką „Pagalba“.

Rinkėjo pažymėjimas būtinas balsuojančiojo identifikavimui. Tai rinkėjui suteikia teisę balsuoti. *Terminalas*, atpažinęs balsuojantįjį bei gavęs leidimą balsuoti, balsuojančiajam pateikia kandidatų sąrašą. Pasirinkus galimą(-us) kandidatą(-us), rinkėjo paklausiama ar patvirtina savo balsą, ir gavus teigiamą atsakymą balsas išsaugomas, priešingu atveju rinkėjas vėl grįžta prie kandidatų pasirinkimo sąrašo.

Internetinėmis bei mobiliosiomis priemonėmis veikiantys balsavimo *terminalai* turi užtikrinti tokio paties lygio reikalavimus bei garantuoti siunčiamų duomenų saugumą.

3.1.2. Projektuojamo objekto konceptuali schema

Balsavimo terminalo konceptuali schema

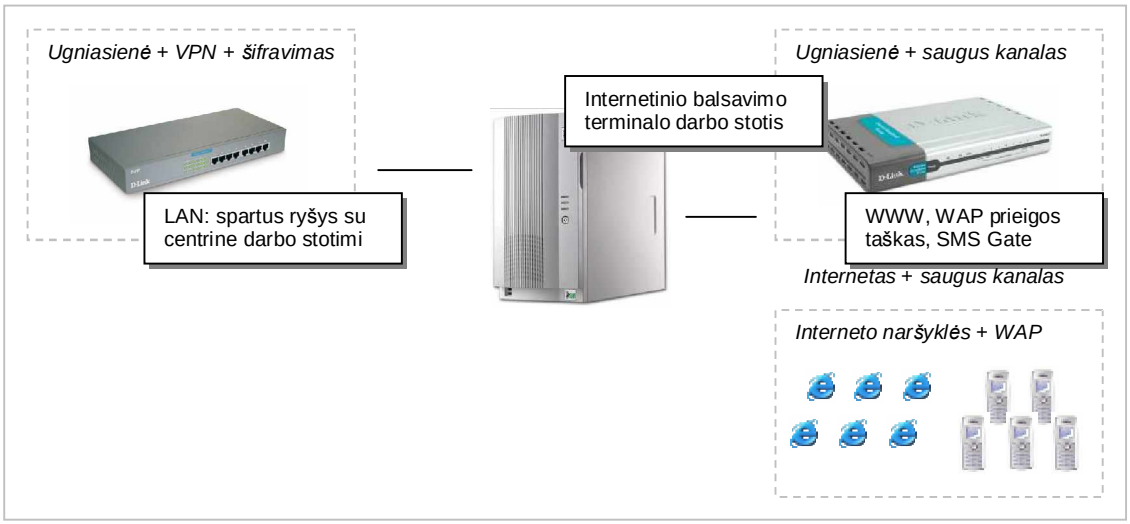


7 pav. Balsavimo terminalo konceptuali schema

Mobilusis balsavimo terminalas turi specialias jungtis, prie kurių prijungiama bevielio ryšio įranga, USB atminties modulis su operacine sistema ir reikalinga programine įranga ir „SmartCard“ tipo kortelių skaitymo/rašymo įrenginys atsarginėms duomenų (balsavimo rezultatų) kopijoms. Įjungus terminalą automatiškai nustatoma sistema - iš atminties modulio pakraunama programinė įranga identifikuoja techninę įrangą, įdiegia reikalingas tvarkykles bei nustato įrengimus. Specialiomis komandomis užmezgamas ryšys ir patikrinamas kanalo pralaidumas. Sėkmingai atlikus procedūras paleidžiama balsavimo programa, kuri pilnai pradeda veikti tik nustatytą rinkimų valandą bei gavusi leidimą iš apygardos darbo stoties.

Per specialius nuskaitymo įrengimus vartotojas identifikuojamas – tam reikalinga asmens tapatybės kortelė bei rinkėjo pažymėjimas. Naudojimosi instrukcijos pateikiamos terminalo ekrane garsinėmis bei vaizdinėmis priemonėmis. Operacijos atliekamos liečiant jautrų ekrano paviršių (esant galimybei terminalas aprūpinamas papildoma pirštų antspaudų atpažinimo įranga).

Internetinio balsavimo terminalo konceptuali schema



8 pav. Internetinio balsavimo terminalo konceptuali schema

Internetinio balsavimo veikimo principai ir reikalavimai identiški mobiliajam balsavimo terminalui, skiriasi tik techninė bei programinė įranga. Stotis prijungta prie dviejų maršrutizatorių, atskirtų techninėmis ir programinėmis priemonėmis. Rinkimų metu rinkėjai gali prisijungti prie specialaus rinkimams skirto tinklapio ir įvedę prisijungimo duomenis atlikti balsavimo procedūrą. Identifikavimas realizuojamas elektroninio parašo priemonėmis (arba analoginėmis) taip užtikrinant asmens tapatumą.

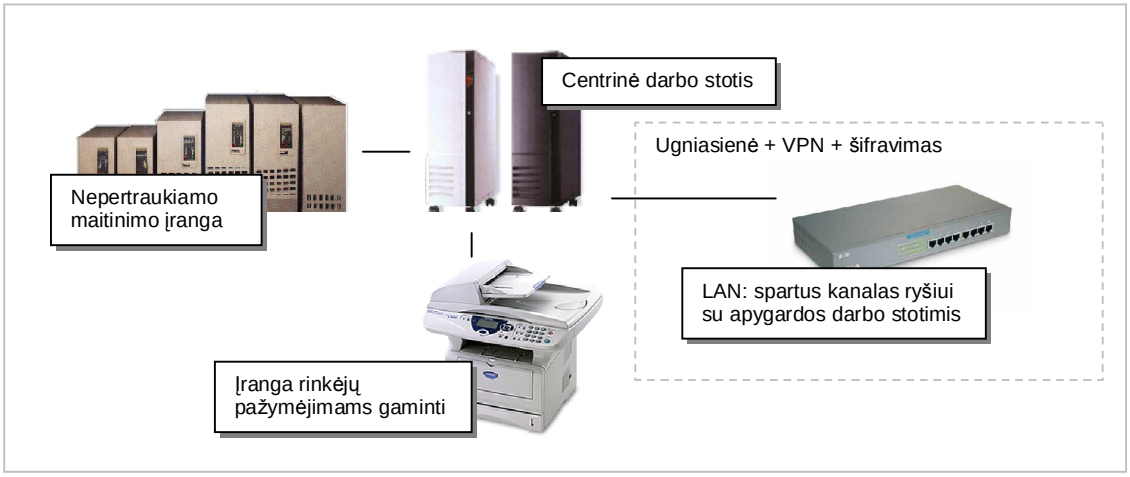
Apygardos darbo stoties konceptuali schema

9 pav. Apygardos darbo stoties konceptuali schema

Rinkimų dieną ar keletą dienų prieš, naudojantis instrukcijomis personalas sukomplektuoja techninę įrangą, stotyje paleidžia specialią automatizuotą programinę įrangą, kuri identifikuoja techninius resursus, įdiegia reikalingas tvarkykles bei nustato specialius tinklo parametrus. Sėkmingai atlikus diegimą patikrinami ryšio kanalai. Numatytu metu apygardos darbo stotis jungiasi prie centrinės darbo stoties, parsisiunčia reikalingus duomenis bei atidaro prieigos kanalą mobiliosioms darbo stotims. Rinkimų metu tarpininkauja siunčiant statistinius duomenis, o vėliau ir rezultatus.

Esant reikalui ir galimybei papildomai prijungiamas balsavimo terminalas balsuoti turintiems teisę, bet negavusiems rinkėjo pažymėjimo.

Centrinės darbo stoties konceptuali schema



10 pav. Centrinės darbo stoties konceptuali schema

Centrinė darbo stotis – tai galingas serverių masyvas, užtikrinantis nepertraukiamą darbą duomenų perdavimą bei stebintis tinklo būklę. Siekiant užtikrinti pastovų darbą energiją stočiai tiekia nepertraukiamo maitinimo šaltiniai visą rinkimų laiką. Stotis prijungta prie didelio pralaidumo interneto prieigos taško bei apsaugota ugniasiene. Priėjimas apygardų darbo stotims suteikiamas tik rinkimų metu.

Stotyje diegiama speciali programinė įranga. Prasidėjus rinkimams gavus specialias užklausas siunčiami duomenys, priimama tinklo būsenos informacija, stebimos galimos klaidos tinkle, taip užtikrinant rinkimų eigos saugumą. Atsitikus nenumatytiems atvejams programinė įranga specialiais pranešimais išveda detalią informaciją apie gedimą ar trikdžius. Tokiu būdu aptarnaujantis personalas gali nedelsiant imtis priemonių ir šalinti gedimus.

Darbo stotis aprūpinta ir technine įranga gaminančia rinkėjų pažymėjimus (esant galimybėms šią funkciją derėtų perduoti specialiai spaustuvei).

3.1.3. Darbo vietų sąsajų su specifikuotomis funkcijomis lentelė

Lentelė Nr. 2

Darbo vieta	Specifikuota funkcija
Centrinė darbo stotis	Balsuojančiųjų duomenų apdorojimas; Duomenų apygardoms siuntimas; Rinkimų eigos stebėjimas; Duomenų surinkimas bei apdorojimas; Rezultatų skaičiavimas.
Apygardos darbo stotis	Rinkėjų duomenų siuntimas; Kandidatų duomenų siuntimas; Statistikos duomenų apdorojimas; Rinkimų eigos stebėjimas; Rinkėjų pažymėjimų spausdinimas; Tarpininkavimas surenkant rinkimų rezultatus;
Balsavimo terminalas	Patogios vartotojo sąsajos užtikrinimas; Saugumo reikalavimų užtikrinimas; Rinkėjo identifikavimas; Rinkėjo balso priėmimas bei apdorojimas;

3.1.4. Kompiuterių darbo vietoms parinkimas ir pagrindimas

Centrinė darbo stotis:

- Ne mažiau 1 GB atminties dideliam duomenų kiekio talpinimui (vienu metu vykdomų procesų gausa privalo būti vykdoma greičiausioje įmanomoje aplinkoje);
- Šiuolaikinis Pentium ar RISC architektūros procesorius (priklausomai nuo pasirinktos operacinės sistemos) galintis greitai apdoroti didelius informacijos kiekius (tai galėtų būti naujausias XEON tipo procesorius, vienu metu sugebantis atlikti bene didžiausią kiekį skaičiavimų);
- Ne mažiau 100 GB disko vietos (disko vieta reikalinga neribotam duomenų apdorojimui užtikrinti);
- SVGA tipo vaizdo plokštė (vaizdo plokštei nėra keliami aukšti reikalavimai, kadangi nėra numatomas darbas su vaizdais);
- USB tipo prievadai, standartinė klaviatūra ir pelė;
- Šiuolaikinis spalvinis vaizduoklis (kadangi dirbama ne su vaizdine medžiaga, vaizduoklio kokybė bendriausiu atveju turėtų tenkinti ergonominius poreikius);
- Kokybiška ir patikima 100/1000Mbit/s greičio tinklo plokštė (kadangi numatomas didelis duomenų srautas, turime užtikrinti didelį duomenų pralaidumą);
- Nepertraukiamo maitinimo šaltiniai galintys aprūpinti elektros energija darbo stotis visą rinkimų laiką.

Sistema privalo nuolat būti stebima, duomenys realiu laiku dubliuojami į keletą (mažiausiai 3) atsarginių serverių, taip išvengiant trikdžių esant darbo stoties gedimams. Esant reikalui turi būti numatyta galimybė nesudėtingam papildomų stočių pajungimui. Bendru atveju darbo stoties kompiuteriai privalo būti sertifikuoti ir pagaminti iš nepriekaištingų komponenčių. [Išsamesnis aprašymas pridedamas priede Nr. 9]

Apygardos darbo stotis:

- Ne mažiau 512 MB atminties (didelis atminties kiekis užtikrins greičiausią realiu laiku užduočių įvykdymą);
- Šiuolaikinis Pentium ar RISC architektūros procesorius (priklausomai nuo operacinės sistemos);
- Ne mažiau 10 GB disko vietos (laikiniems duomenims saugoti);
- Galimybė realiu laiku duomenis bei procesus dubliuoti lygiagrečiai dirbančiomis analogiškomis stotimis;
- SVGA tipo vaizdo plokštė;
- USB tipo prievadai, standartinė klaviatūra bei pelė;
- Spalvotas vaizduoklis;
- Ne mažiau 10/100Mbit/s greičio tinklo plokštė (kokybiška plokštė, užtikrinanti pastovų ryšį);
- Nepertraukiamo maitinimo šaltinis užtikrinantis stoties darbą visą rinkimų laiką.

Kadangi sistema tik laikinai dirba su kritiniais duomenimis, svarbiausia yra užtikrinti pastovų ryšį su centrine darbo stotimi bei rinkimų terminalais. [Išsamesnis aprašymas pridedamas priede Nr. 9]

Terminalas:

- 128 MB atminties (atminties kiekis turi užtikrinti minimalius reikalavimus);
- Šiuolaikinis Pentium ar RISC architektūros procesorius;
- Ne mažiau 1GB disko vietos saugoti rezultatams (diskas turi būti apsaugotas nuo galimų išorinių poveikių);
- USB tipo prievadai;
- Ne mažiau 10/100Mbit/s greičio tinklo plokštė (kokybiška plokštė, užtikrinanti pastovų ryšį);
- SVGA tipo vaizdo plokštė (plokštė turi atitikti minimalius reikalavimus);
- Prisilietimams jautrus ekranas ar kitas įvedimo įrenginys;
- „BarCode“ nuskaitymo įranga (esant galimybei prijungiama ir pirštų antspaudų atpažinimo įranga);
- Autonominis elektros tiekimas užtikrinantis stoties veikimą visą rinkimų laiką;

- Mechaninis sprendimas rinkimų rezultatams fiksuoti.

Terminalinė stotis privalo būti izoliuota nuo išorės veiksnių, įvedimo įrenginiai saugiai atskirti nuo sisteminių plokščių. Išoriškai stotis gali skirtis nuo standartinių komponenčių, kadangi saugumo reikalavimų tenkinimas gali pareikalauti netradicinių sprendimų. [Išsamesnis aprašymas pridedamas priede Nr. 10]

3.1.5. Aparatinės priemonės

Normaliam darbui organizuoti nepakaks vien tik kompiuterių – būtina specifinė duomenų įvedimo įranga. Patogų darbą garantuos vaizduokliai su jautriu prisilietimam ekranu. Rinkėjui identifikuoti reikalinga „BarCode“ nuskaitymo įranga, o esant galimybei ir pirštų antspaudų atpažinimo prietaisas. Ryšio užtikrinimui naudojama dviejų tipų įranga – bevielio tinklo skirstytuvai su stiprinimo antenomis, padėsiantys sujungti mobiliuosius terminalus į tinklą, bei „LAN“ tinklo prieigos įranga. [Išsamesni aprašymai pridedami prieduose Nr. 11, Nr. 12, Nr. 13, Nr. 14]

3.1.6. Programinės priemonės

Galimos/rekomenduojamos operacinės sistemos:

- Microsoft Windows 2000/XP;
- Sun Solaris;
- Red Hat arba SuSE Linux.

Operacinės sistemos pasirinkimą įtakoja gamintojų atsakomybė į galimus padarinius naudojimosi metu. Galimas variantas, jog apygardų bei terminalinių stočių valdymui bus realizuotas atskiras operacinis paketas remiantis vienomis ar kitomis technologijomis. Siekiant paprastumo, realus ir OS CD sprendimas (operacinė sistema užkraunama iš kompaktinio disko kartu su reikalinga programine įranga bei iš anksto numatytais nustatymais) arba kitoks, CD skaitymo nereikalaujantis sprendimas.

Papildoma programinė įranga:

- Java™ RE (siekiant užtikrinti daugelio operacinių sistemų palaikymo);
- Programinis atitinkamos darbo stoties valdymo sprendimas.

Atsižvelgiant į daugelį pašalinių veiksnių galimas ir kitas programinės įrangos variantas, tačiau turi būti užtikrinama programų bei aplinkos kokybė, o taip pat lankstumas.

Galimos duomenų bazės:

- Oracle 9i;

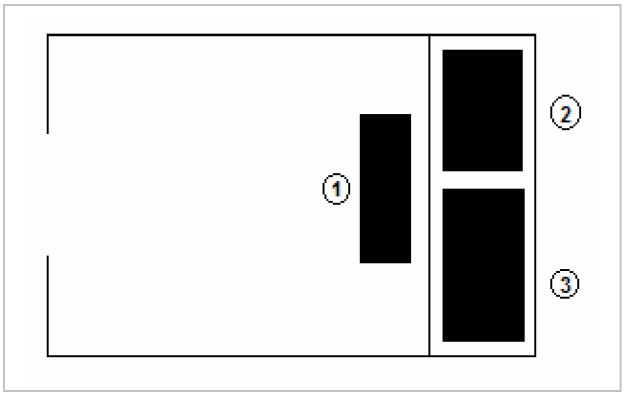
- PostgreSQL;
- IBM DB2.

Duomenų bazės panaudojimas būtinas tik centrinėje darbo stotyje, kadangi būtina užtikrinti operatyvų duomenų apdorojimą bei užtikrinti duomenų kokybę. Programinės įrangos pasirinkimą įtakos gamintojo politika į galimą problemų šalinimą bei atsakomybę. Turi būti užtikrinamas pilnas palaikymas bei priežiūra.

Apygardų bei terminalinėse stotyse, atsižvelgiant į duomenų svarbumą bei kritiškumą, turėtų būti naudojami specialūs įrašų failai ar kitokie sprendimai siekiant išlaikyti nepriklausomumą nuo pašalinių programinių paketų, užtikrinant informacijos saugumą bei atkūrimą.

3.1.7. Patalpų projektas su darbo vietų ir kitos aparatūros išdėstymu

Mobiliojo balsavimo terminalo schema



11 pav. Mobiliojo balsavimo terminalo schema

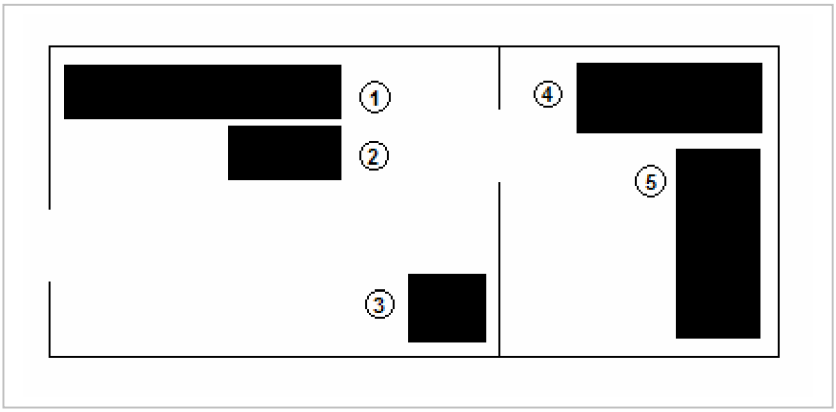
Schemoje pavaizduotų objektų sąrašas:

1. Rinkėjo balso įvedimo įrenginys;
2. Nepertraukiamo maitinimo įranga;
3. Terminalinė įranga.

Balsavimo terminalas turi atitikti visus ergonomikos reikalavimus, patogus ir nepavojingas balsuojančiojo sveikatai. Terminalinė ir maitinimo įranga privalo būti atskirta ir izoliuota nuo galimo paveikimo tiek iš

balsavietės vidaus tiek ir iš išorės. Izoliuojanti įranga turi apsaugoti nuo magnetinio poveikio, mikrobangų ir kitokio poveikio, galinčio vienaip ir kitaip įtakoti rezultatams. Privaloma galimybė valdyti įvedimo įrenginio aukštį ir padėtį, siekiant suteikti galimybę balsuoti neįgaliesiems.

Apygardos darbo vietų schema



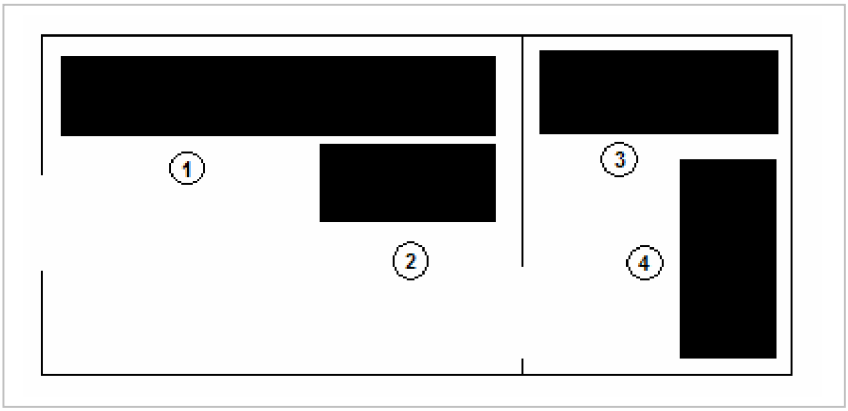
12 pav. Apygardos darbo vietų schema

Schemoje pavaizduotų objektų sąrašas:

1. Apygardos komisija;
2. Komisijos techninė įranga;
3. Balsavimo terminalas;
4. Apygardos darbo stotis;
5. Stoties nepertraukiamo maitinimo šaltinis.

Darbo stotis ir maitinimo įranga patalpinta specialioje saugykloje, kuri atitinka visus gamintojo reikalaujamus normatyvus (temperatūra, oro drėgnumas ir panašiai), be to turi būti užtikrinta apsauga nuo pašalinių asmenų. Balsavimo terminalas turi užtikrinti balsuojančiojo konfidencialumą. Balsavimo patalpoje turi būti iškabinti Vyriausiosios rinkimų komisijos išleisti rinkimų įsakymai ir kiti aktai. Iš balsavimo patalpos, perėjimo į ją patalpų (koridorių) ir 50 metrų atstumu aplink pastatą, kuriame yra balsavimo patalpa, turi būti pašalinta rinkimų agitacijos medžiaga, išskyrus tą, kurią išleido Vyriausioji rinkimų komisija. Taip pat turi būti paruoštos darbo vietos rinkimų komisijos nariams, vietos rinkimų stebėtojams. Parengta balsavimo patalpa uždaroma, antspauduojama, perduodama saugoti policijai ir apie tai apylinkės rinkimų komisijos pirmininkas praneša miesto, rajono rinkimų komisijai.

Centrinės rinkimų komisijos darbo vieta



13 pav. Centrinės rinkimų komisijos darbo vieta

Schemoje pavaizduotų objektų sąrašas:

1. Rinkimų komisijos darbo vieta;
2. Darbo bei stebėjimo techninė įranga;
3. Nepertraukiamo maitinimo įranga;
4. Centrinė darbo stotis.

Esminiai reikalavimai nesiskiria nuo reikalavimų apygardos darbo vietos reikalavimų – privalo būti užtikrinama patogi aplinka dirbantiesiems, techninė įranga patalpinta specialius reikalavimus atitinkančioje saugykloje.

Schemos ir paminėti reikalavimai yra tik rekomendacinio pobūdžio. Realiomis sąlygomis dėl vienokių ar kitokių apribojimų tokio išdėstymo realizuoti gali būti neįmanoma. Bet kokiu atveju turi būti tenkinami bent įstatymų numatyti reikalavimai.

3.1.7. Techninės tinklo priemonės

Tarp centrinės ir rinkiminės apygardos darbo stočių:

Visa rinkimų informacija turi būti perduodama taip, kad pašaliniai asmenys, negalėtų stebėti ir pakeisti duomenų. Norint tai užtikrinti, reikia, kad tinklu keliaujanti informacija būtų koduojama. Tai pasiekti galima naudojant VPN – Virtualus Privatus Tinklas (Virtual Private Network) dėka. Tai atskirų, nutolusių kompiuterinių tinklų sujungimas per Internetą. VPN tinklo technologija leidžia saugiai ir greitai sujungti skirtingose vietose esančius kompiuterius į vieną tinklą, pasinaudojant interneto infrastruktūra. VPN tinklai kitas technologijas lenkia savo įdiegimo spartumu, lankstumu ir ekonomine nauda - mokestis už VPN ryšį visada bus mažesnis nei mokestis už skirtinę liniją, reikalingą vienam padaliniui prijungti prie centrinės

būstinės. Lieka tik klausimas, kokiu būdu VPN sujungimą realizuoti, t. y. programinėmis ar aparatinėmis priemonėmis.

VPN realizacija naudojant komutatorius:

Privalumai:

- Tinkle gali būti kompiuteriai su skirtingomis operacinėmis sistemomis;
- Jungiant kompiuterių grupes, reikia nustatyti pagrindinį maršrutizatorių.

Trūkumai:

- Kainos atžvilgiu neapsimoka jungti į VPN vieną kompiuterį tinkle su kitu kompiuteriu.

VPN programinė realizacija:

Privalumai:

- Nereikalinga papildoma aparatinė įranga;
- Paprastas nustatymas;
- Patogu sujungti trumpam laikui du ir daugiau kompiuterių.

Trūkumai:

- Visur turi būti to paties kūrėjo operacinės sistemos.

VPN sujungimas bus reikalingas tarp centrinės darbo stoties ir apygardos darbo stoties, be to tai bus reikalinga gana trumpa laiką. Todėl VPN sujungimą reiktų realizuoti programinėmis priemonėmis. Tam reikalinga įdiegti daug VPN susijungimų palaikančią programinę įrangą centrinėje darbo stotyje ir VPN klientines dalis apygardos bei terminalinėse darbo stotyse. Prie centrinės darbo stoties jungsis daugiau nei 70 apygardų darbo stočių ir joms kartu reiks išsiųsti didelius informacijos srautus, todėl būtina darbo stotis prijungti prie aukštą interneto pralaidumą turinčio interneto tiekėjo. Centrinės darbo stoties ir apygardos kompiuteriuose naudojamos 10/100/1000 MB tinklo
plokštės. Mobilieji terminalai tarpusavyje ir su apygardos darbo stotimi apjungiami bevielio tinklo įranga palaikančia saugų ryšį. Signalo stiprumui palaikyti turi būti panaudotos daugiakryptės antenos.

3.1.8. Tinklo operacinės sistemos parinkimas, pagrindimas

Darbo stotyse galima naudoti tiek Microsoft kompanijos, tiek Linux operacines sistemas. Pasirinkimas turi būti tiek teisiškai, tiek ir finansiškai pasvertas, kadangi galimas operacinės sistemos klaidas turėtų šalinti programinės įrangos tiekėjas.

Linux operacinės sistemos pasirinkimas galėtų būti realus dėl programinio kodo statuso, galimybės diegti reikalingus ir šalinti nereikalingus sistemos modulius, platus reikalingos programinės įrangos pasirinkimas bei galimybės realizuoti USB BOOT funkciją (operacinė sistema paleidžiama iš USB atminties modulio). Visgi versijų gausa bei programinio kodo atvirumas verčia suabejoti dėl patikimumo.

Microsoft Windows 2000/XP programinės įrangos patikimumą užtikrina pati kompanija, todėl dėl operatyvaus klaidų šalinimo bei savo srities atsakomybės prisiėmimas gali būti esminiai renkantis operacinę sistemą. Tačiau paketo kaina gali įtakoti šį pasirinkimą.

Terminalinių darbo stočių programinės įrangos pasirinkimą apsprendžia sisteminio sprendimo pasirinkimas. Atsižvelgiant į paprasto diegimo pageidavimus bei saugumą reikalingas nestandartinis sprendimas. Tai galėtų būti kokia nors modifikuota Linux versija, įdėmiai peržiūrėta ir išnagrinėta šia sritį išmanančių specialistų.

Atsižvelgiant į dabartinę sprendimų įvairovę optimalus operacinių sistemų pasirinkimo variantas būtų toks:

- Centrinė darbo stotis – Windows 2000 Server;
- Apygardos darbo stotis – Windows 2000 Server;
- Mobilusis balsavimo terminalas – RedHat Linux.

Windows operacinė sistema pasirinkta dėl įprasto ir patogaus vartojimo, pakankamo saugumo ir programinės tinklo realizacijos įvairovės. Balsavimo terminalams RedHat Linux naudojama dėl turtingos programinės įrangos pasirinkimo bei aparatinės sistemos valdiklių gausos.

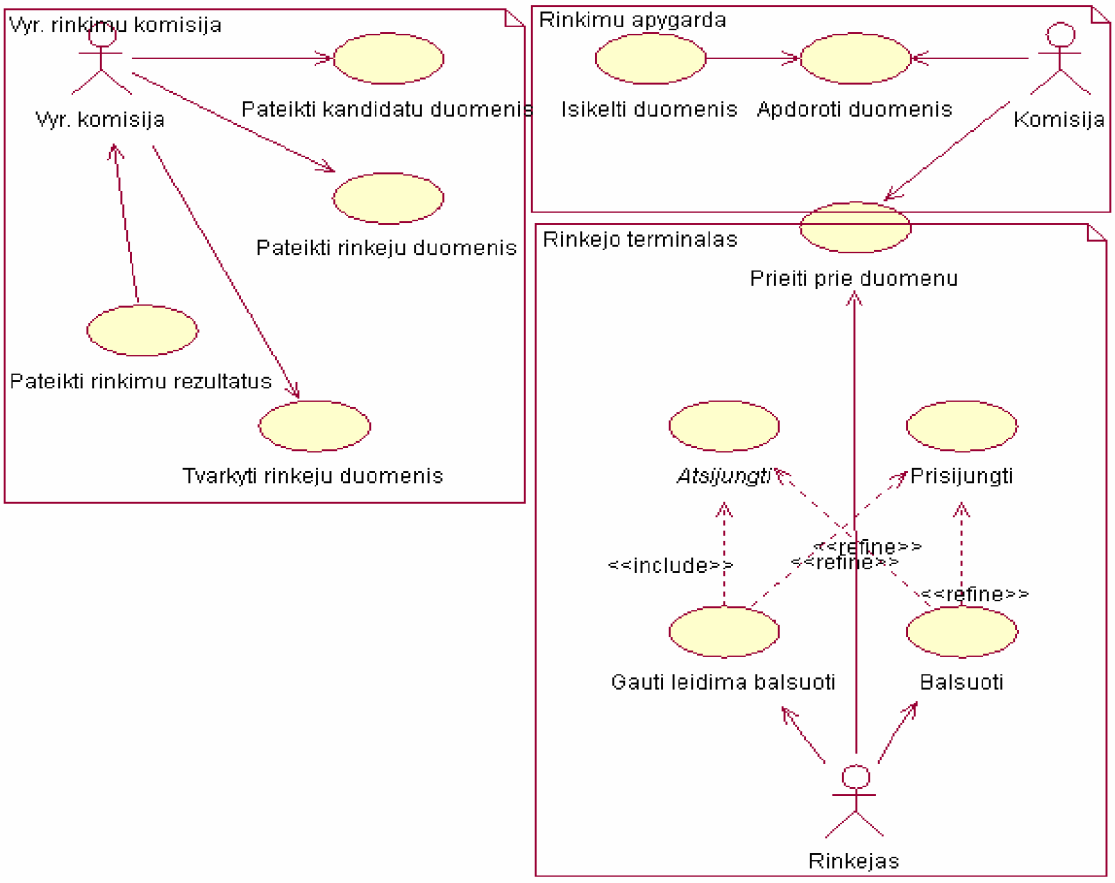
Siūloma programinė įranga realizavimo metu gali keistis dėl galimo gamintojų sisteminio palaikymo nutraukimo.

3.2. Reikalavimų modelis

3.2.1. Vartotojų panaudojimo atvejų diagrama

Panaudojimo atvejis yra sistemos elgsenos vienetas – vartotojo ir sistemos sąveikų seka, kuri duoda vartotojui reikšmingą rezultatą. Kiekvieną panaudojimo atvejį aprašo įvykių srautas, kuris apibrėžia, ką sistema turi padaryti jos panaudojimo metu.

Diagrama padės išsivaizduoti galimus panaudojimo atvejus tarp tarpusavyje bendradarbiaujančių sistemų, su kuriomis dirba atitinkami žmonės. Atskirų diagramų Apygardos, Mobiliajam balsavimo terminalams ar kitiems nereikia, kadangi bendras algoritmas yra vienodas.



14 pav. Vartotojų panaudojimų atvejų diagrama

Pagrindiniai aktoriai:

- Vyr. komisija - daugiausiai teisių turintis vartotojas, galintis matyti rinkimų duomenis (rezultatus, balsavusiųjų kiekį), koreguoti rinkėjų duomenis, stebėti rinkimų proceso eigą.
- Komisija – gali peržiūrėti apygardos turimą informaciją, su tam tikrais apribojimais ją koreguoti, stebėti rinkimų proceso eigą.

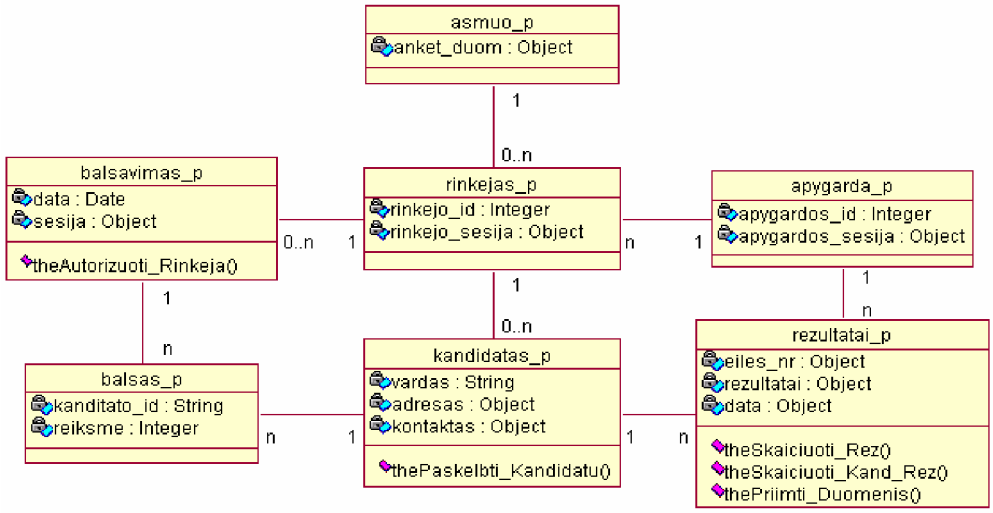
- Rinkėjas – mažiausiai teisių turinti vartotojas, kuris gauna leidimą balsuoti ir turi teisę atiduoti savo balsą už norimą kandidatą (-us).

Pagrindiniai panaudojimo atvejai:

- Pateikti kandidatų duomenis – gali atlikti tik *Vyr. komisija*.
- Tvarkyti duomenis – gali atlikti tik *Vyr. komisija*.
- Naudotis apygardos DB - gali atlikti *Komisija.*
- Pateikti rinkimų rezultatus – gali atlikti *Komisija*.
- Balsuoti – gali atlikti *Rinkėjas*.
- Gauti leidimą balsuoti – gali atlikti tik *Rinkėjas*.

3.2.2. Dalykinės srities klasių diagrama

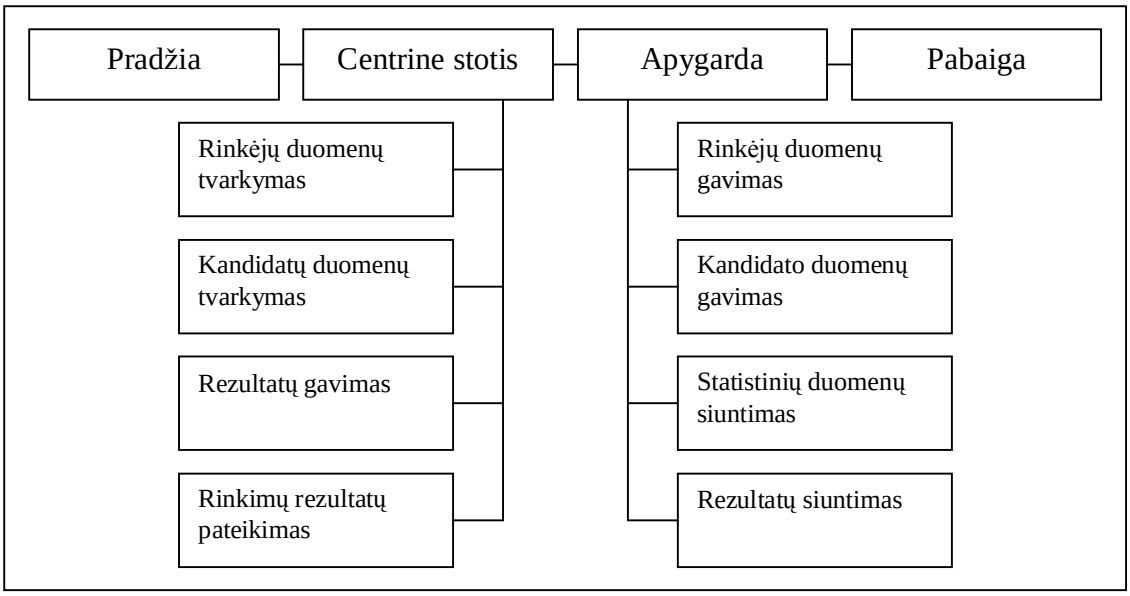
Klasių diagrama rodo klases ir jų tarpusavio ryšius. Klasė – tai objektų aibė, kurios elementai turi vienodą struktūrą, elgsena, ryšius ir semantiką. Klasės struktūrines savybes apibrėžia jos atributai. Ryšiai rodo, kad objektai sąveikauja tarpusavyje.



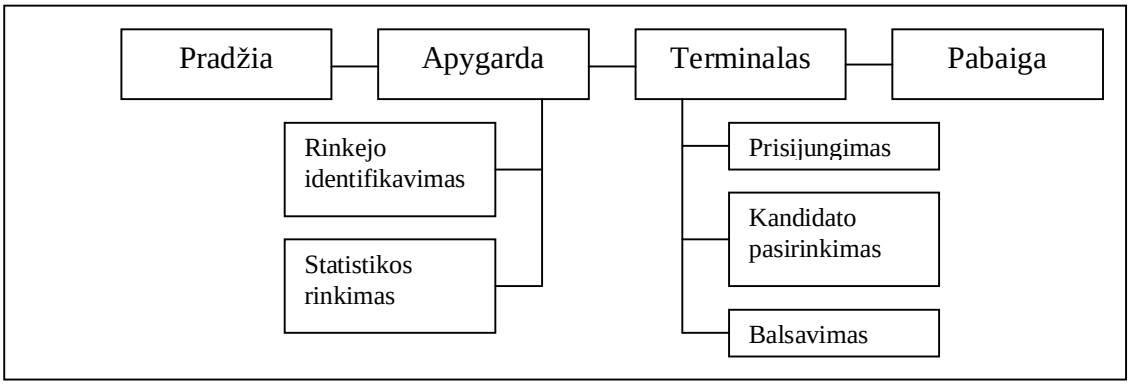
15 pav. Dalykinės srities klasių diagrama

Tai tik bendrinė klasių diagrama pagrindiniam objektų bei jų duomenų modeliui apžvelgti.

3.2.3. Vartotojo interfeiso modelis



16 pav. Interfeiso Centrinė stotis – Apygarda diagrama



17 pav. Interfeiso Apygarda – Balsavimo Terminalas diagrama

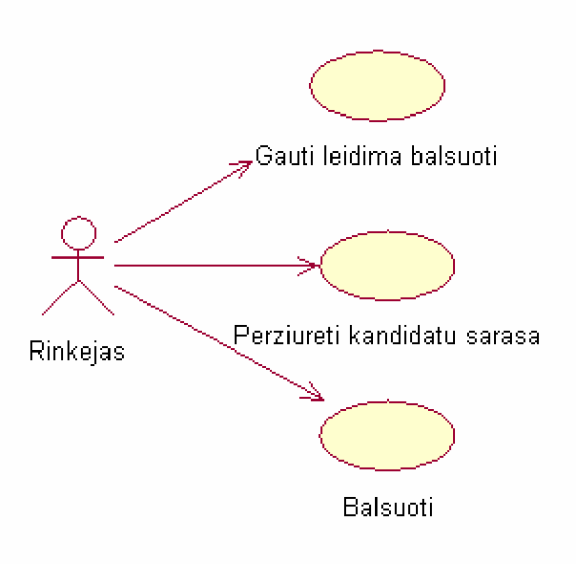
3.3. Sistemos projektas

3.3.1. Projekto tikslas

Suprojektuoti ir realizuoti analizės dalyje išnagrinėtą mobiliųjų e-rinkimų sistemą.

3.3.2. Sistemos panaudojimo atvejų diagramos

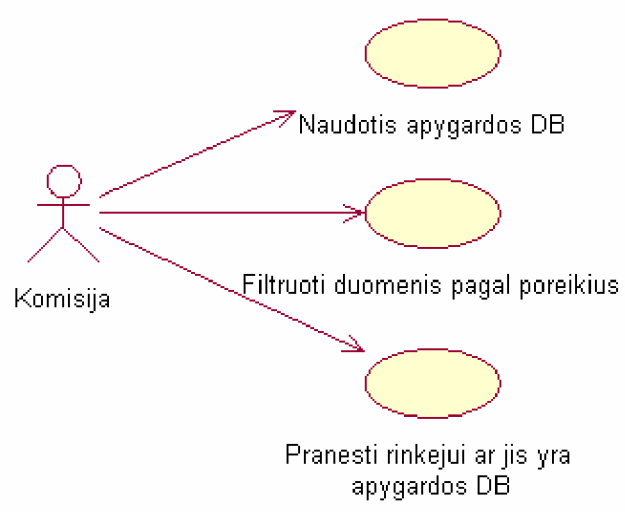
Rinkėjo panaudojimo atvejų diagrama:



18 pav. Rinkėjo panaudojimo atvejų diagrama

“Rinkėjas” yra asmuo kuris turi teisę dalyvauti rinkimuose ir balsuoti už norimą kandidatą. Pirmiausia jis turi „Gauti leidimą balsuoti“, tada gali „Peržiūrėti kandidatų sąrašą“ ir balsuoti už pasirinktą kandidatą.

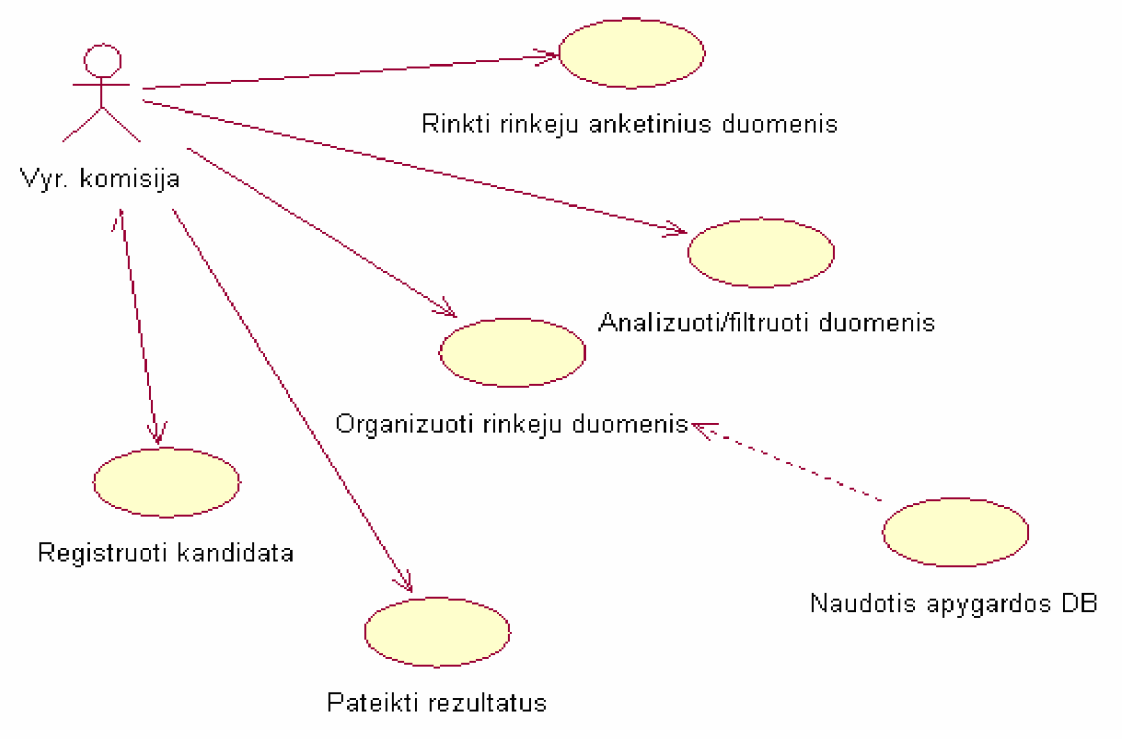
Komisijos panaudojimo atvejų diagrama:



19 pav. Komisijos panaudojimo atvejų diagrama

“Komisija” turi priėjimą prie tai apygardai priklausančių rinkėjų duomenų. Gali filtruoti duomenis pagal rinkėjo asmens kodą taip suteikdami rinkėjui informaciją ar jis yra įtrauktas į apygardos duomenų bazę ir ar turi teisę balsuoti.

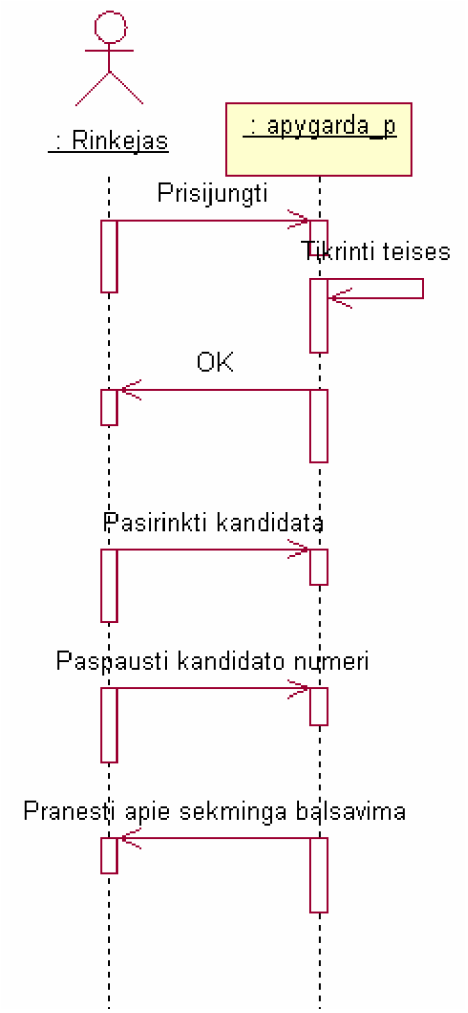
Vyr. komisijos panaudojimo atvejų diagrama:



20 pav. Vyr. komisijos panaudojimo atvejų diagrama

“Vyr. komisija” gali registruoti naują kandidatą, rinkti rinkėjų duomenis į DB (t.y. išrinkti tuos kurie turi teisę balsuoti, yra sulaukę 18 m. ir t.t.). Taip pat „Vyr. Komisija“ gali analizuoti duomenis pagal poreikius, bei naudotis visų apygardų duomenimis. Viena pagrindinių funkcijų – periodiškas rinkimų rezultatų pateikimas.

3.3.3. Panaudojimo atvejų scenarijų diagramos

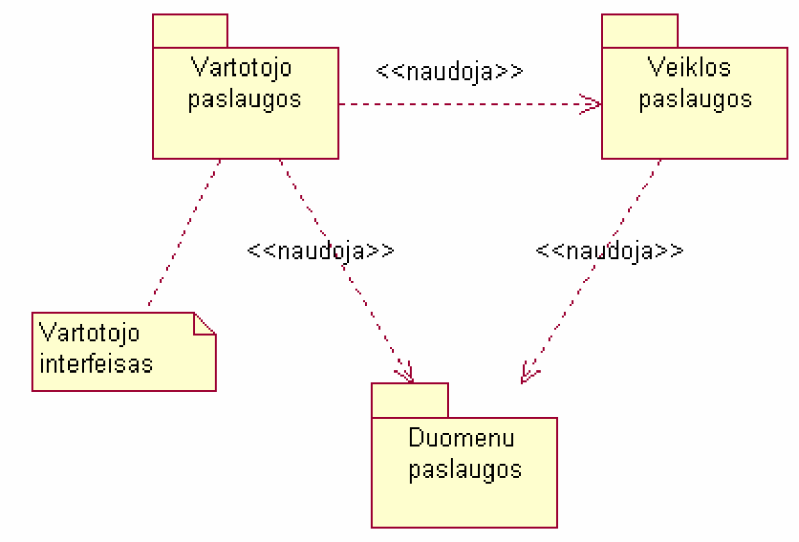


21 pav. Rinkėjo sisteminės dalies sekų diagrama

Rinkėjas eidamas balsuoti prie terminalo turi atsinešti rinkimų biuletenį su atspausdintu BAR kodu, kurį gavo paštu, atsispausdino iš interneto arba gavo iš apygardos rinkimų komisijos.

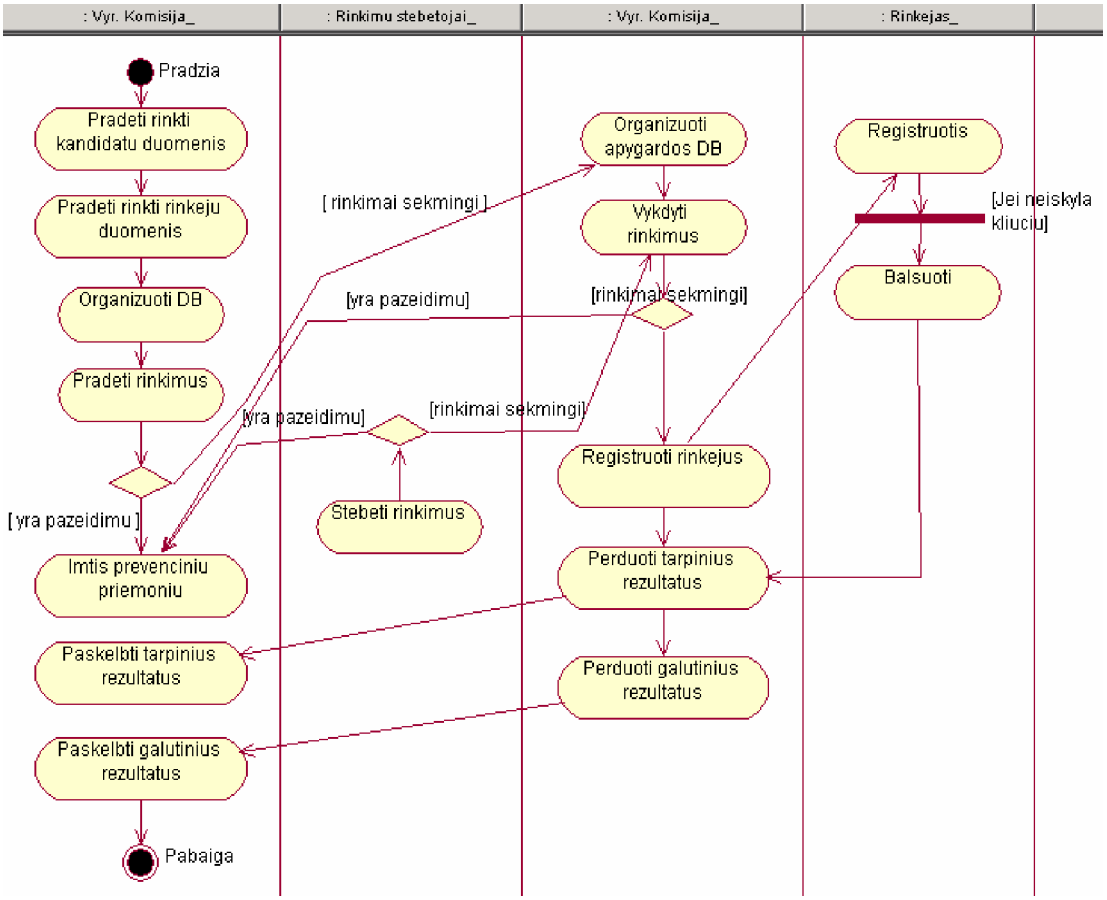
BAR kodas nuskaitomas ir tikrinama ar toks vartotojas yra apygardos DB ir ar jis dar nebalsavęs. Jei rinkėjas turi teisę balsuoti jam pateikiamas kandidatų sąrašas. Išsirinkęs kandidatą rinkėjas balsuoja ir duodamas pranešimas apie sėkmingą balsavimą.

3.3.4. Sistemos architektūros modelis



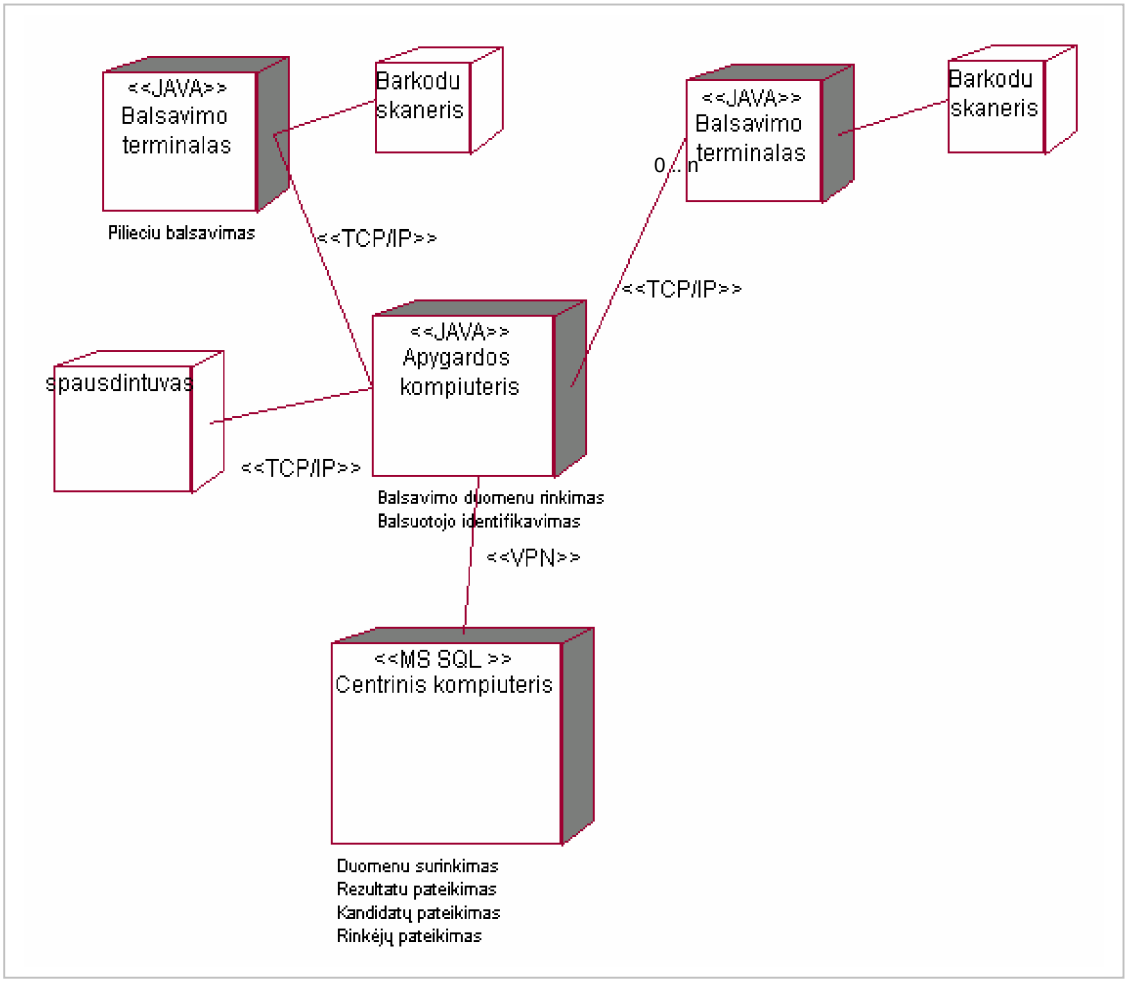
22 pav. Sistemos architektūros modelis

3.3.5. Sistemos veiklos modelis



23 pav. Sistemos veiklos modelis

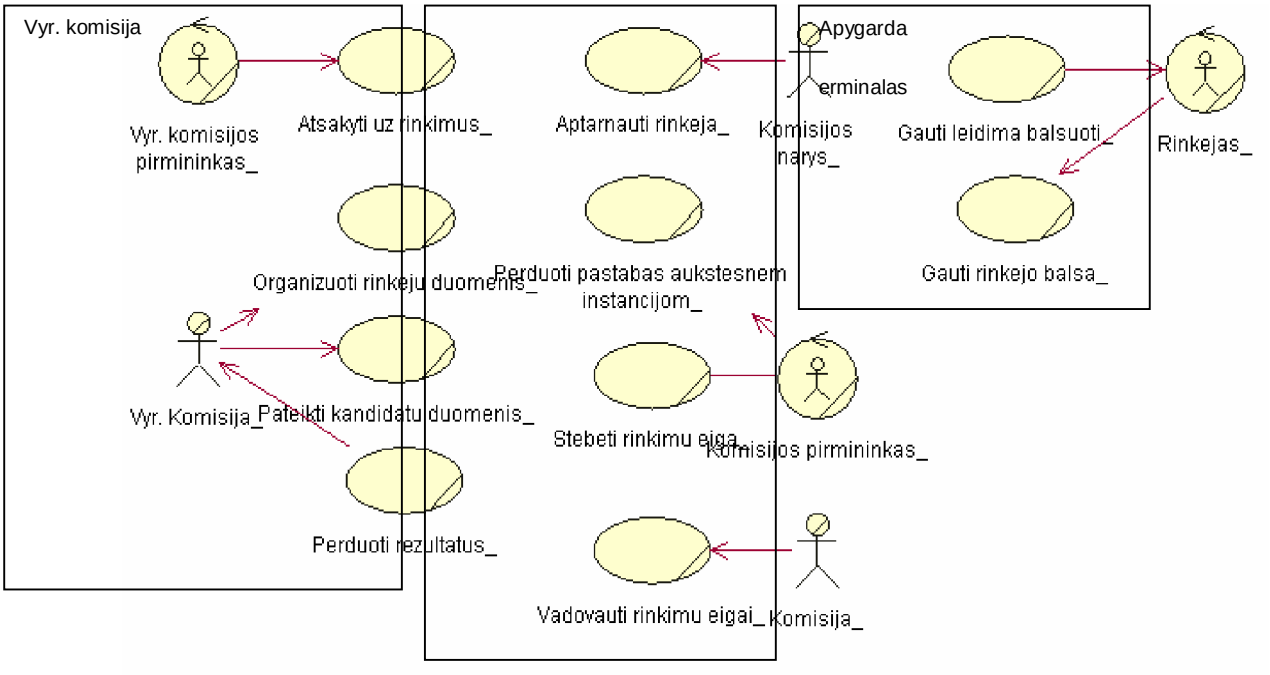
3.3.6. Realizacijos modelis



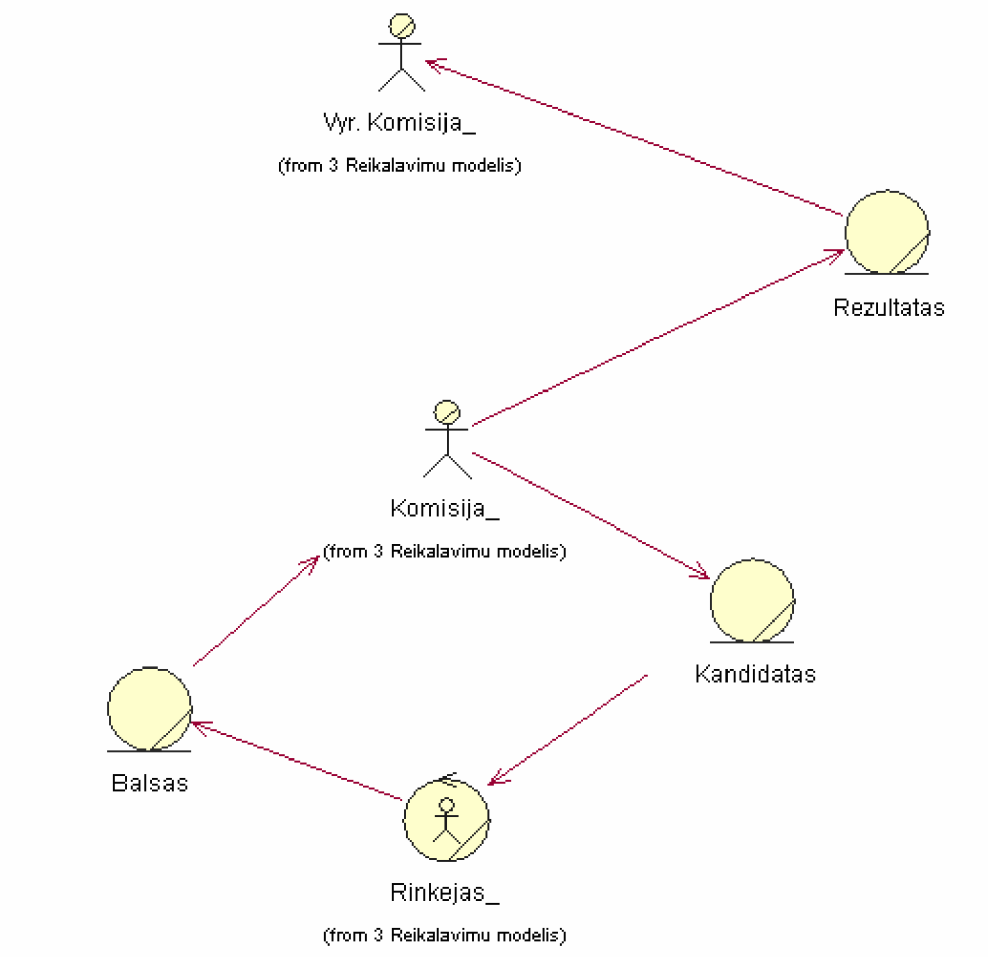
24 pav. Paskirstymo diagrama

Dirbti su sistema vartotojai galės paleidę ją. Rinkėjams sistema jau bus paleista taip pat bus pateiktos nuorodos kokie balsavimo žingsniai.

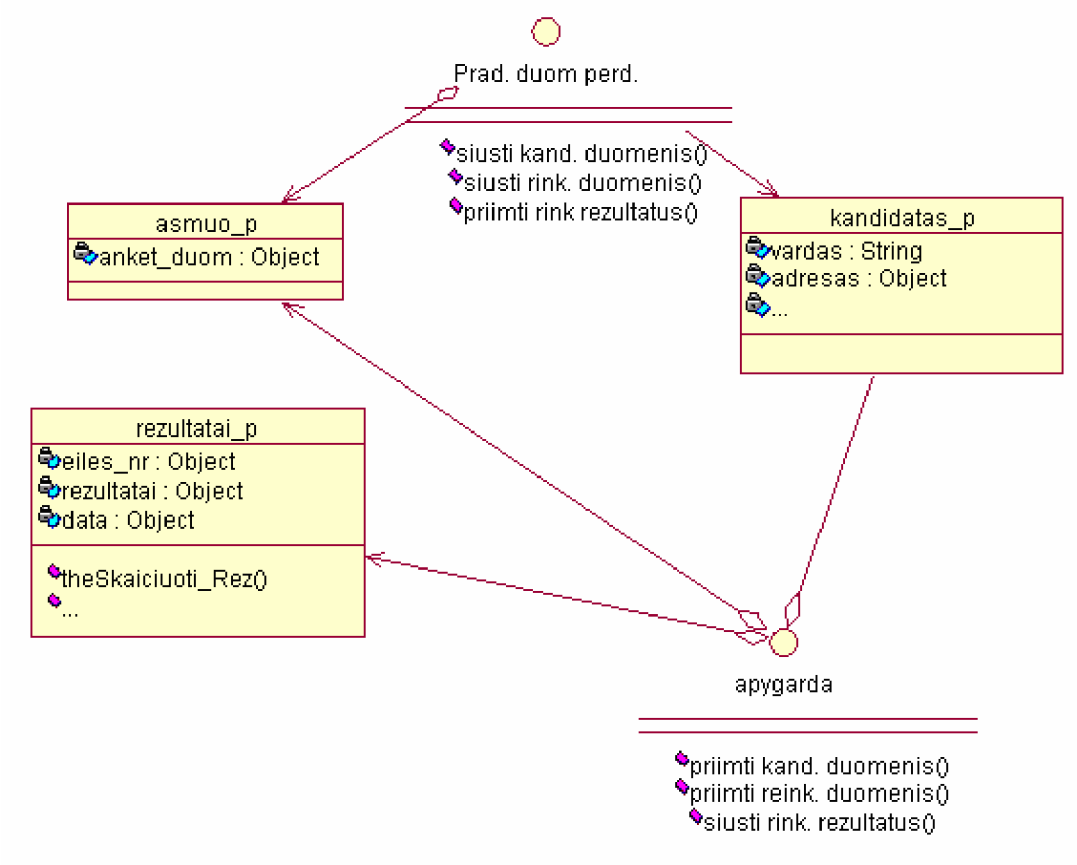
3.3.7. Komponentinis sistemos modelis



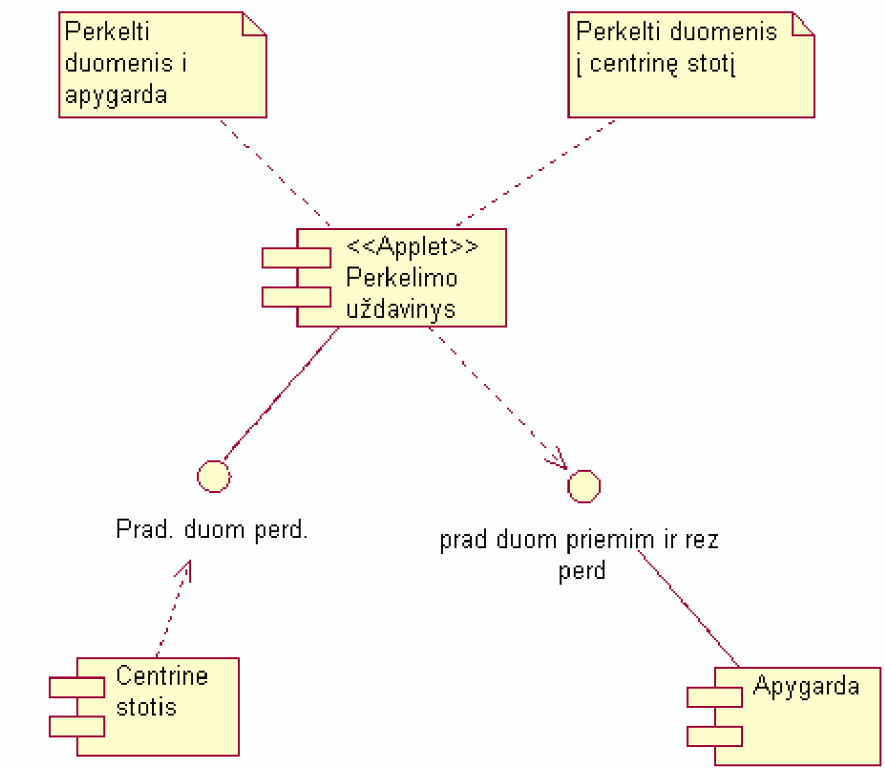
25 pav. Veiklos panaudojimo atvejų diagrama



26 pav. Duomenų perdavimo veiklos konceptų modelis



27 pav. Veiklos tipų modelis



28 pav. Programinės realizacijos architektūra

3.3.8. Duomenų bazės modelis

Duomenų bazės modelis pateikiamas priede Nr. 14.

3.2. Projekto išvados

Sistema gana sudėtinga ir reikalaujanti netradicinių sprendimų. Pirmiausia turi būti pasirinktas optimalus aparatinis sprendimas, nuo kurio priklauso ir daugelis programinio realizavimo procedūrų.

Techninio projektavimo dalyje išnagrinėti optimalūs variantai bei reikalavimai centrinei darbo stočiai. Microsoft kompanijos operacinės sistemos (Windows) ir tinkliniai sprendimai supaprastina informacijos valdymą. Taip pat šie sprendimai tenkina sistemos reikalavimus, lengvai adaptuojami ir integruojami. Kompanijos specialistų patirtis užtikrins pilnavertę priežiūrą veikimo metu, o programiniai sprendimai padės išsaugoti duomenų bei procesų lygiagretiškumą bei saugumą.

Apygardų darbo stotims daugeliu atvejų pilnai pakaktų modifikuotos Linux versijos, kadangi didžioji dalis darbų – informacijos mainai, sistemos stebėjimas, nereikalauja ypatingų priemonių, be to daugelis jų privalės būti realizuota pačios rinkimų programinės įrangos.

Programiniai sprendimai, nesant techniniams apribojimams, gana nesudėtingai realizuoja pageidaujamus funkcionalumus – duomenų perdavimas, informacijos teisingumo užtikrinimas, sinchronizavimas, monitoringas ir kt. Visgi atsižvelgiant į saugumo reikalavimus, realizacija turėtų užtikrinti paprastumą ir lengvą klaidų šalinimą. Tokiu būdu bus užtikrinama kodo kokybė bei nesunkiai aptinkamos ir ištaisomos klaidos.

4. Eksperimentinis tyrimas

4.1. Eksperimentinė projekto dalis

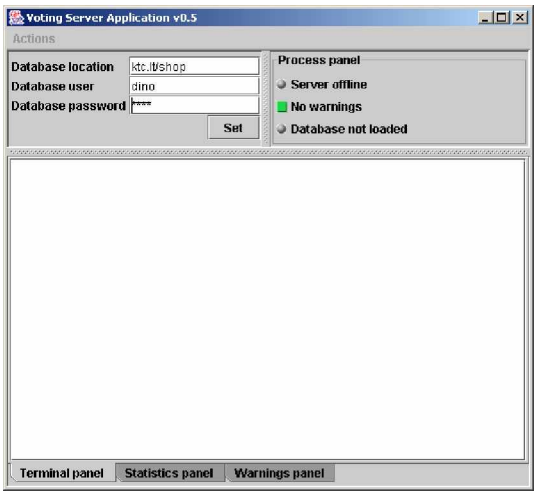
Realizuotas eksperimentinis projekto modelis demonstruoja, kaip realiai turi atrodyti ir veikti mobiliųjų E-rinkimų taikomoji sistema.

Eksperimentinėje dalyje realizuota:

- Duomenų bazė
- Centrinė darbo stotis;
- Centrinės darbo stoties vartotojo sąsaja;
- Apygardos darbo stotis;
- Apygardos darbo stoties vartotojo sąsaja;
- Terminalas;
- Terminalo vartotojo sąsaja;

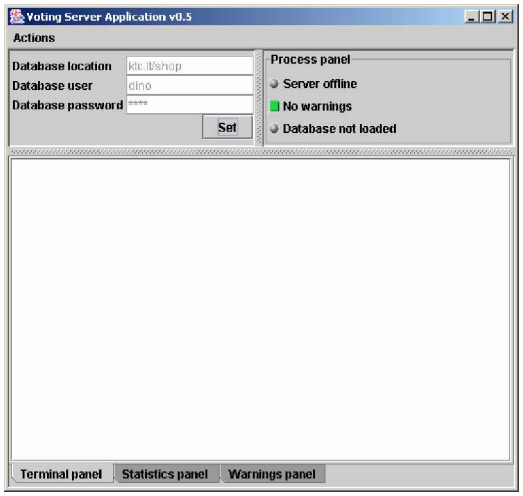
Centrinė darbo stotis

Centrinės darbo stoties vartotojo sąsajos meniu punktai valdo ir paleidžia centrinės darbo stoties servisus (aplikacijas). Pagrindinis vartotojo sąsajos langas, kuris atsidaro paleidus programą (29 pav.).



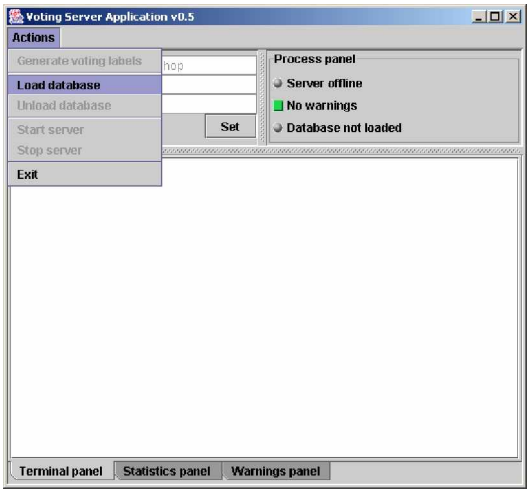
29 pav. Centrinės darbo stoties vartotojos sąsajos pagrindinis langas

Prieš vykdant centrinės darbo stoties pagrindinio serviso paleidimą reikia įvesti laukų „Database location“, „Database user“, „Database password“ reikšmes. Įvedus teisingas reikšmes spaudžiame mygtuką „Set“, kad aktyvuotųsi meniu punktas „Actions“ (30 pav.).



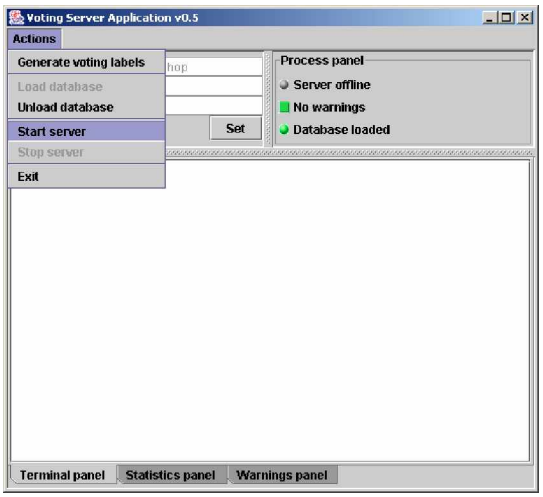
30 pav. Centrinės darbo stoties meniu punkto „Actions“ aktyvavimas

Sekančiu veiksmu vykdome susijungimą su duomenų baze spausdami „Actions“ → „Load database“ (31 pav.). Duomenų bazė paleista.



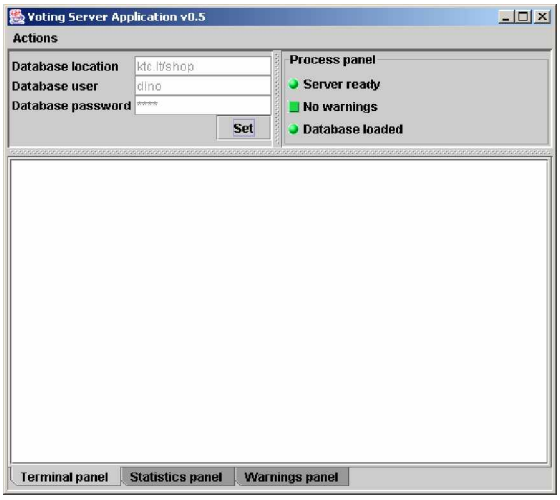
31 pav. Centrinės darbo stoties duomenų bazės paleidimas

Paleidus duomenų bazę būsenos indikatorius „Database not loaded“ keičia būseną į „Database loaded“ ir indikatoriaus spalva pasikeičia iš pilkos į žalią. Taip pat aktyvuojasi meniu punktas „Start server“ (32 pav.).



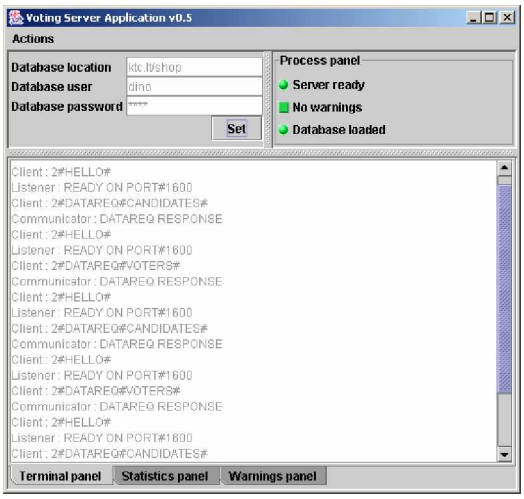
32 pav. Duomenų bazė paleista, vykdomas pagrindinių procesų paleidimas

Spausdami „Start server“ paleidžiame pagrindinius centrinės darbo stoties procesus. Darbo stoties būsenos indikatorius „Server offline“ keičia reikšmę į „Server online“, indikatoriaus spalva keičiasi iš pilkos į žalią, darbo stotis paleista (33 pav.).



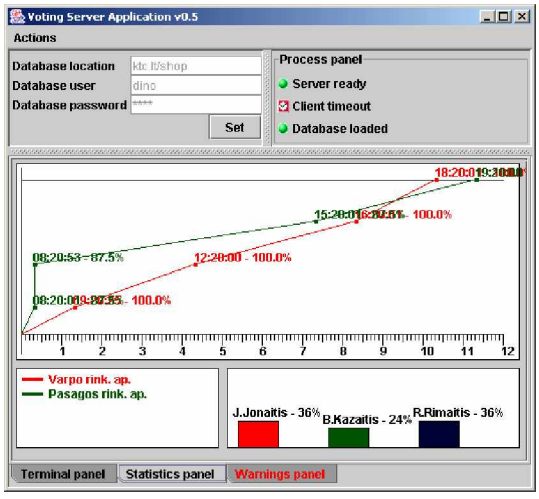
33 pav. Centrinė darbo stotis pilnai paleista, galima pradėti darbą

Centrinė darbo stotis sekančiais veiksmais apdoroja apygardos siunčiamas užklausas ir siunčia atsakymus.



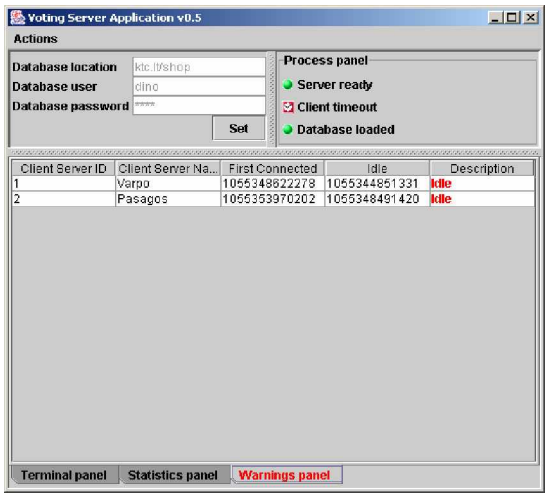
34 pav. Užklausų priėmimas ir atsakymų siuntimas

Taip pat periodiškai priima statistinius duomenis iš apygardos darbo stoties. Statistiniai duomenys atvaizduojami panelėje „Statistics panel“ (34 pav.). Pagrindinėje diagramoje vaizduojami statistiniai duomenys. Apatinės dalies kairėje pusėje išvedamos apygardos, kurių statistika atvaizduota, bei jų skiriamosios spalvos. Rinkimams pasibaigus centrinė darbo stotis gauna iš apygardos darbo stoties rinkimų rezultatus, juos apdoroja ir atvaizduoja stulpeline diagrama darbalaukio apačioje, dešinėje pusėje. Apygardos būsenos indikatorius „Client offline“ rodo, kad apygardos darbo stotis yra atsijungusi, kai siunčiami statistiniai duomenys ar rezultatai šis indikatorius keičia reikšmę į „Client online“ (35 pav.).



35 pav. Centrinė darbo stoties rinkimų statistikos duomenys

Paskutinė panelė „Warnings panel“ Joje pateikiama informacija apie apygardas, apygardų prisijungimo būsena, numeriai, pirmo prisijungimo laikas ir kiek laiko apygarda yra atsijungus nuo paskutinio atsijungimo laiko (36 pav.)

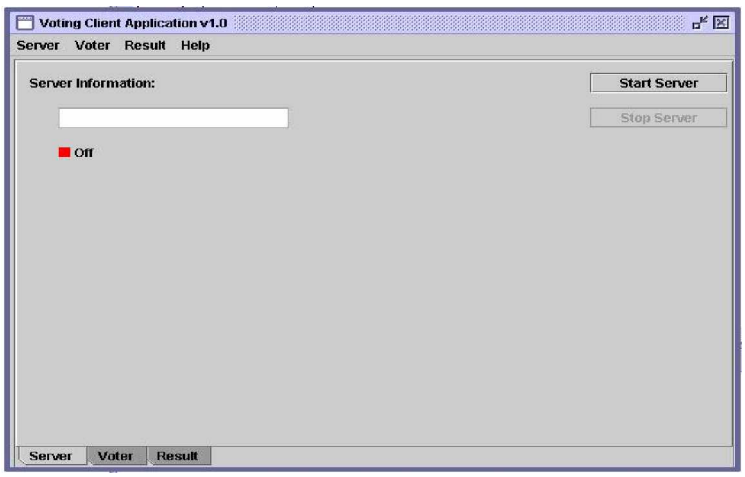


36 pav. Centrinė darbo stoties rinkimų „Warning panel“ duomenys

***A**pygardos darbo stotis*

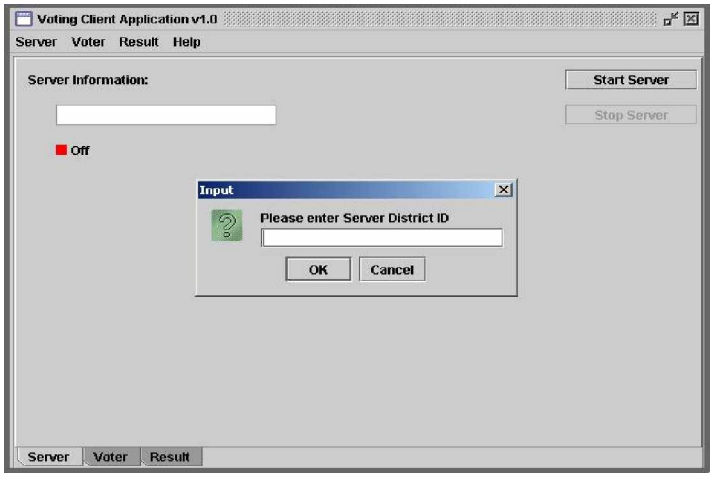
Apygardos darbo stoties vartotojo sąsajos meniu punktai valdo ir paleidžia atitinkamus apygardos darbo stoties servisus (aplikacijas).

Pati apygardos darbo stoties vartotojo sąsaja paleidžiama paspaudus ant jos piktogramos (ikonos). Pagrindinis langas, kuris atsidaro paleidus vartotojo sąsajos programą (37 pav.).



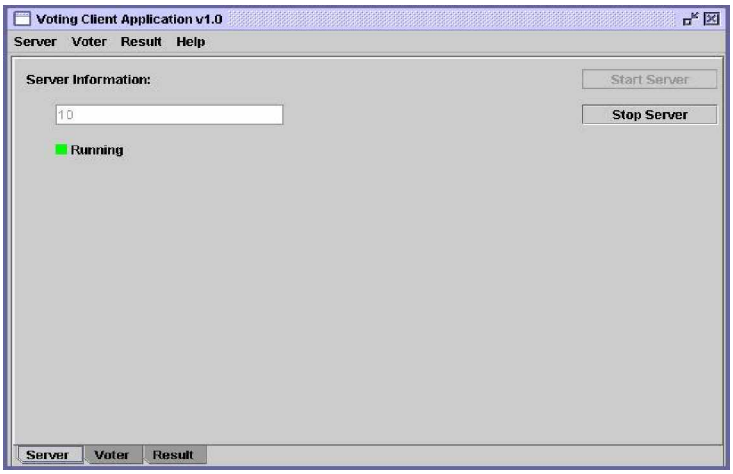
37 pav. Apygardos darbo stoties vartotojos sąsajos pagrindinis langas

Vartotojo sąsają galima suskirstyti į tris dalis – tai meniu punktai išsidėstę horizontaliai lango viršutinėje dalyje, pagrindinė panelė, kurioje matoma visa informacija ir katalogo struktūra horizontaliai išsidėstę meniu punktai esantys apatinėje lango dalyje, kurių pagalbą galima pasirinkti vieną iš trijų skirtingų langų. Pagrindinis langas yra skirtas apygardos darbo stoties pagrindinio serviso paleidimui. Pasirinkus meniu punktą "Server"→"Start server" arba panelyje paspaudus mygtuką "Start server" pradedamas apygardos darbo stoties darbas, prieš tai įvedus apygardos darbo stoties numerį "Server District ID" (38 pav.).



38 pav. Konkrečios rinkimų apygardos pasirinkimo langas

Teisingai įvedus egzistuojantį rinkimų apygardos numerį paleidžiamas pagrindinis apygardos darbo stoties servisas, pradedamas darbo stoties darbas. "Server" lango panelėje galime pamatyti pagrindinio serviso paleidimo būseną – jei langelyje dega raudona šviesa vadinasi pagrindinis servisas yra išjungtas. Įjungus pagrindinį servisą langelyje išsižiebia žalia šviesa (39 pav.).

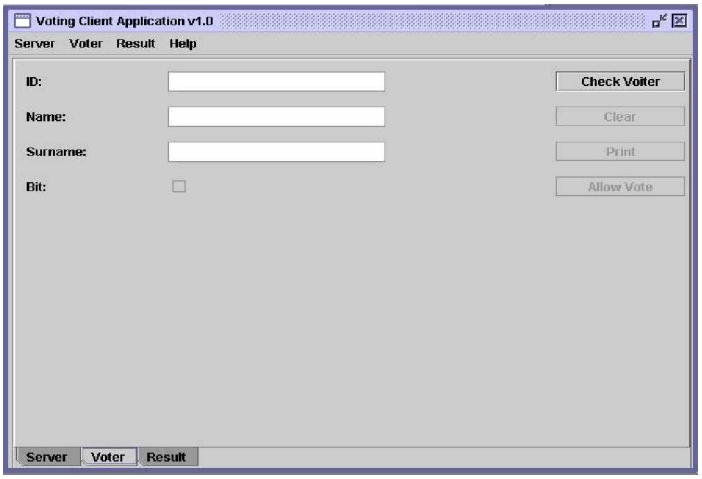


39 pav. Langelyje deganti žalia šviesa informuoja, kad pagrindinis servisas paleistas

Jei norime sustabdyti apygardos darbo stoties pagrindinį servisą reikia spausti “Stop server” mygtuką pagrindinėje panelėje arba “Server”→”Stop server” meniu punktą.

Pagrindinis apygardos darbo stoties servisas valdo visus kitus servisus, todėl jį būtina paleisti, jei norime vykdyti sekančias komandas. Paleidus apygardos darbo stotį, ji automatiškai siunčia užklausas centrinei darbo stočiai ir gauna iš centrinės darbo stoties rinkėjų bei kandidatų sąrašus. Jeigu centrinė darbo stotis tuo metu negali vykdyti susijungimo, tuomet apygardos darbo stotis inicijuoja susijungimą po tam tikro laiko tarpo. Sekantis žingsnis - rinkėjo tapatybės nustatymas.

Apygardos darbo stoties administratorius (rinkimų apygardos pirmininkas), pagal LR piliečio pase arba identifikavimo kortelėje esančią nuotrauką nustato ar žmogus atėjo balsuoti su savo dokumentais. Jei viskas tvarkoj, administratorius pasirenka katalogo struktūros meniu “voter” skirsnį (40 pav.).

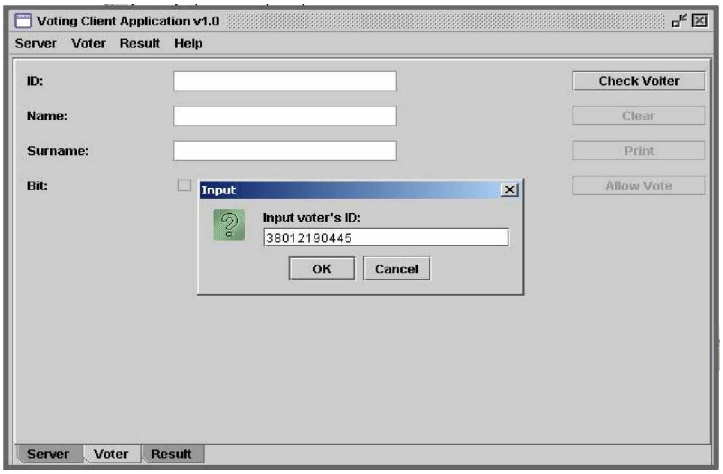


40 pav. Langas rinkėjo tikrinimui

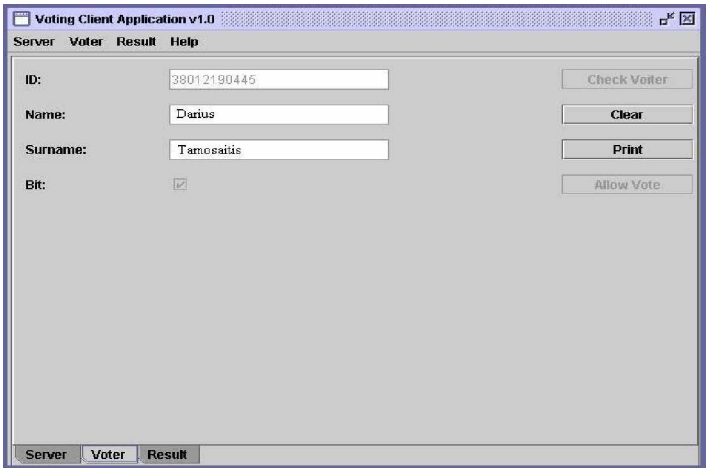
Tada pagrindinėje panelėje arba meniu punkte “Voter” reikia paspausti “Check Voter” mygtuką ir į atsidariusį langą įvesti 11 skaitmenų asmens kodą (41 pav.). Įvedus neteisingus duomenis procedūra reikia pakartoti iš naujo. Asmens kodas tikrinamas pagal apygardos duomenų bazės įrašus, ir jei randamas identiškas įrašas, įrašui esančiam ‘Asmens kodas’ lauke, tada į panelę išvedami duomenys apie rinkėją – Asmens kodas, Vardas ir Pavardė (42 pav.). Priešingu atveju, jei toks asmens kodas apygardos duomenų bazėje neegzistuoja – į ekraną išvedamas pranešimas, kad nėra tokio rinkėjo kodo.

Jei rinkėjo duomenys išvedami į panelę, vadinasi jis turi teisę balsuoti rinkimuose ir administratorius spaudžia panelyje mygtuką “Allow vote” arba meniu punktą “Voter”→”Allow vote”, jei dar tas laukas nėra užpildytas varnele. Jei išvedus rinkėjo duomenis laukas „Bit“ užpildytas varnele, vadinasi rinkėjas jau užsiregistravęs. Paspaudus “Allow vote” mygtuką apygardos duomenų bazėje į atitinkamą lauką įrašomas vienetukas, kuris reiškia , kad rinkėjas gauna teisę balsuoti. “Print” mygtukas panelyje arba “Voter”→”Print” meniu punktas

numatyti tam atvejui, jei rinkėjas į rinkimus ateina be kvietimo lapelio (pametęs arba negavęs). Jei jis pagal sąrašus yra priskirtas tai apygardai į, kurią atėjo balsuoti (įvedus asmens kodą randami rinkėjo duomenys), jam galima atspausdinti kvietimo lapelio su brūkšniniu kodu kopiją.



41 pav. Langas asmens kodo įvedimui

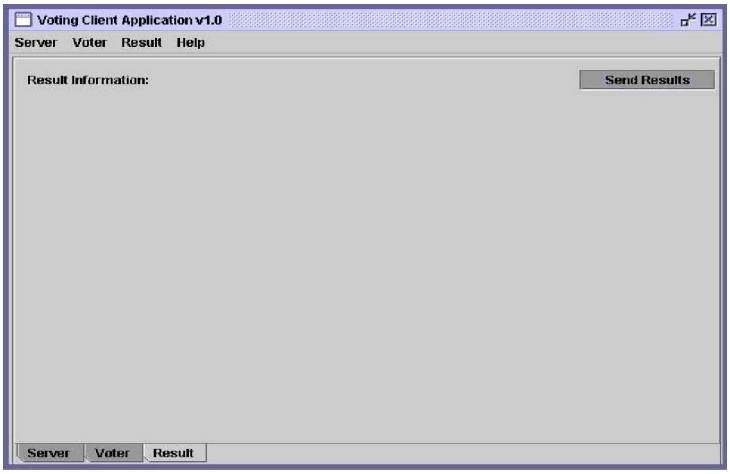


42 pav. Radus rinkėją duomenų bazėje išvedama informacija apie jį, matome kad rinkėjas jau užsiregistravęs balsavimui.

Voter panelyje ir meniu punkte “voter” esantis mygtukas “Clear” leidžia išvalyti iš panelio laukų rinkėjo duomenis, tam kad būtų galima patikrinti kitą rinkėją (nepaspaudus “clear” mygtuko vartotojo sąsaja neaktyvuoja “Check voter” mygtuko). Rinkėjo atpažinimo pagal asmens kodą procedūra kartojama kiekvienam rinkėjui visų rinkimų eigos metu.

Statistikos duomenų formavimas ir siuntimas centrinei darbo stočiai vykdomas automatiškai, kas valandą.

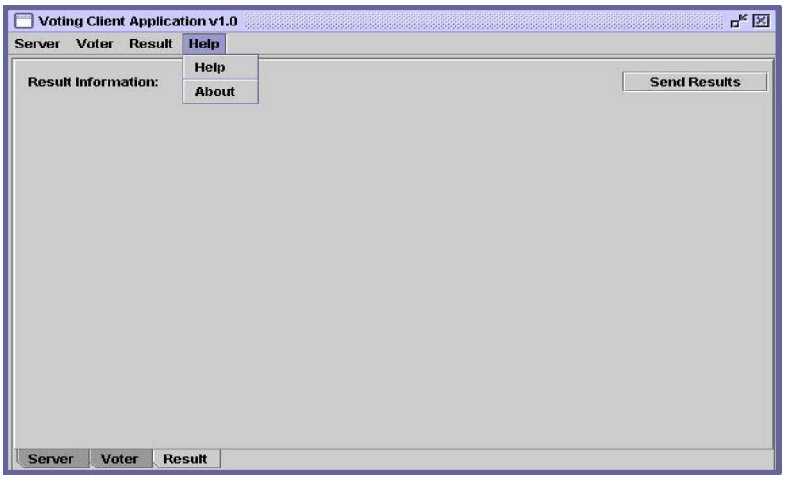
Kitas etapas – rezultatų suformavimas ir siuntimas į centrinę darbo stotį. Pasibaigus rinkimams rinkimų rezultatus būtina išsiųsti į centrinę darbo stotį, galutinių rezultatų formavimui bei pateikimui. Šio serviso langą galima atsidaryti paspaudus į katalogo struktūros meniu “Result” punktą (43 pav.).



43 pav. Rinkimų rezultatų siuntimas

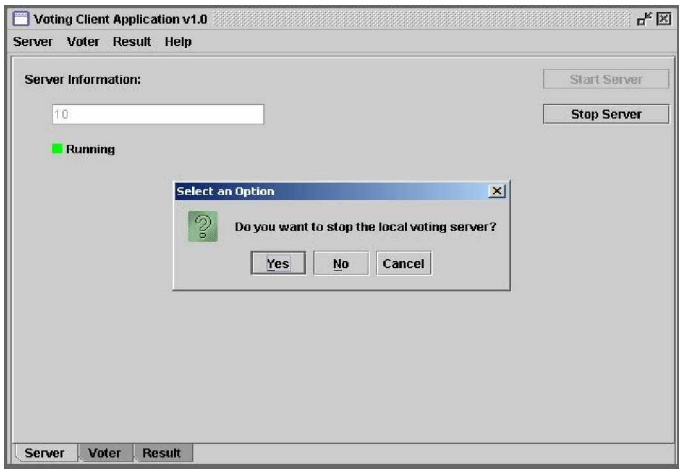
Panelyje arba meniu punkte “Result” paspaudus mygtuką “Send results” paleidžiamas atitinkamas apygardos darbo stoties servisas, kuris išsiunčia rinkimų apygardos balsavimo rezultatus į centrinę darbo stotį.

Šioje vartotojo sąsajoje numatytas ir pagalbos priemonės, kurių padedami vartotojai gali sužinoti kaip naudotis pačia vartotojo sąsaja. Be to galima pažiūrėti, kuri programos versija yra naudojama ir kas jos autorius (44 pav.).



44 pav. Pagalba vartotojui

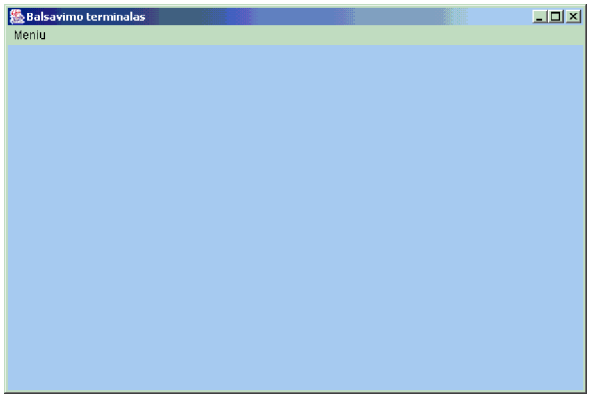
Paskutinis darbas kurį reikia atlikti apygardos darbo stoties administratoriui pasibaigus rinkimams, tai apygardos darbo stoties pagrindinio serviso sustabdymas. Kad tai padaryti, reikia katalogo struktūros meniu pasirinkti “server” punktą tada atsidaro pagrindinis vartotojo sąsajos langas, kuriame nuspaudus mygtuką “Stop server” sustabdoma apygardos darbo stoties pagrindinio serviso veikla (45 pav.).



45 pav. Apygardos darbo stoties pagrindinio serviso išjungimas

Balsavimo terminalas

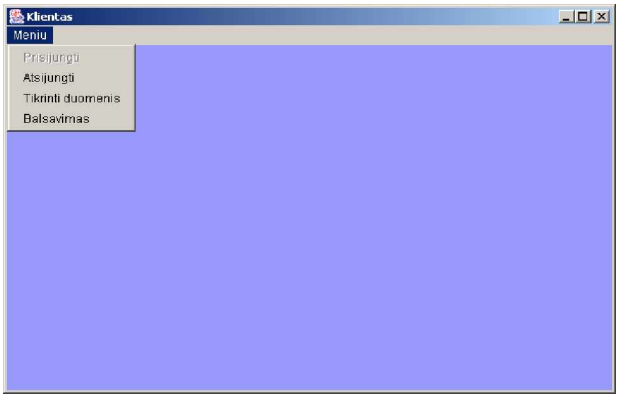
Balsavimo terminalo sąsaja paleidžiama paspaudus ant jos piktogramos (ikonos). Pagrindinis langas, kuris atsidaro paleidus vartotojo sąsajos programą (46 pav.).



46 pav. Balsavimo terminalo sąsajos pagrindinis langas

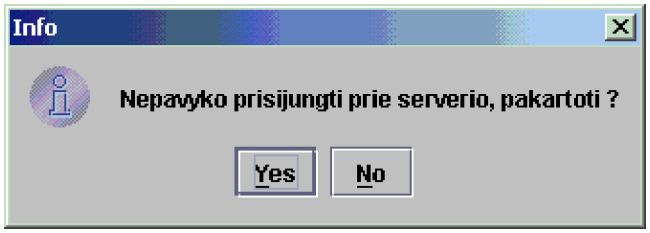
Paspaudę meniu lauką „Meniu“ matome keturis punktus (47 pav.).

- „Prisijungti“ – skirtas prisijungimui prie apygardos darbo stoties, prisijungus šis punktas tampa neaktyvus.
- „Atsijungti“ – skirtas atsijungti nuo apygardos darbo stoties, aktyvus tik po prisijungimo.
- „Tikrinti duomenis“ – prisijungus reikia patikrinti ar apygardos darbo stotis turi reikalingus duomenis, aktyvus tik po prisijungimo.
- „Balsavimas“ – skirtas pereiti iš valdymo rėžimo į balsavimo rėžimą, aktyvus tik kai duomenys yra apygardos darbo stotyje.



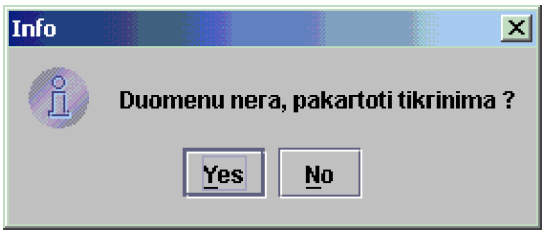
47 pav. Meniu punkto „Meniu“ papunkčiai

Jei prisijungti prie darbo stoties nepavyksta į ekraną yra išvedamas pranešimas (48 pav.).



48 pav. Prisijungti prie apygardos darbo stoties nepavyko

Jei prisijungus prie darbo stoties nepavyksta gauti duomenų į ekraną yra išvedamas pranešimas (49 pav.).

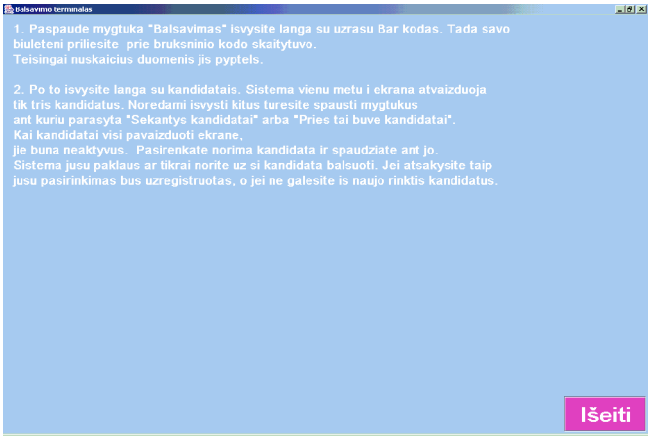


49 pav. Apygardos darbo stotis neturi duomenų reikalingų darbui pradėti

Prisijungus prie apygardos darbo stoties ir nustačius, kad balsavimui reikalingus duomenis apygardos darbo stotis turi, galima pereiti į balsavimo režimą paspaudus meniu punkto „Meniu“ papunktį „Balsavimas“. Tai atlikus atsidaro langas su dviem dideliais mygtukais (50 pav.);

50 pav. Balsavimo pradžia

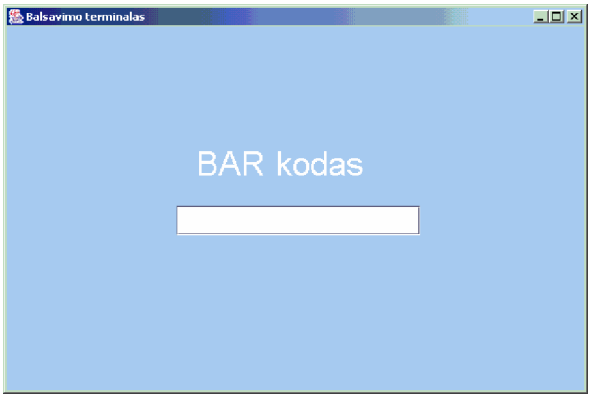
Mygtukas „Pagalba“ – jį paspaudus išvedamas langas, kuriame galima rasti informaciją, kaip elgtis sekančiuose balsavimo etapuose (51 pav.)



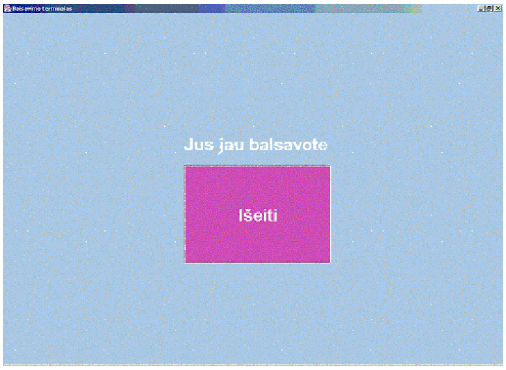
51 pav. Pagalbos langas, informacija apie balsavimo procedūrą

Meniu mygtukas „Balsavimas“ – paspaudus jį patenkama į langą kuriame yra laukas su pavadinimu BAR kodas (51 pav.). Į šį lauką informacija įvedama iš brūkšninio kodo skaitytuvo, kuris atlikęs duomenų įvedimą pypteli. Įvesta informacija – tai rinkėjo identifikacinis numeris (asmens kodas). Sekančiu veiksmu šis rinkėjo identifikacinis (asmens kodas) numeris yra siunčiamas apygardos darbo stočiai, siekiant sužinoti ar

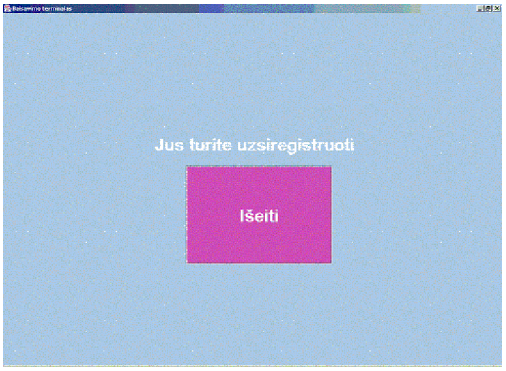
rinkėjas gali balsuoti. Galimi keturi atsakymo variantai : a) Jūs jau balsavote (52 pav.); b) Galite balsuoti c) Jums reikia užsiregistruoti (53 pav.); d) Apie jus nerasta jokiu duomenų (54 pav.). Jei rinkėjas gali balsuoti jis išvysta langą su kandidatų vardais ir pavardėmis (55 pav.);



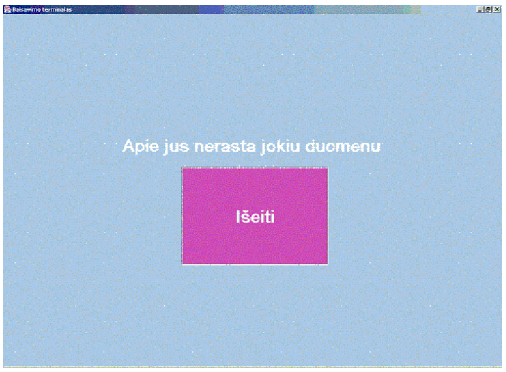
52 pav. Rinkėjo autorizacijos langas



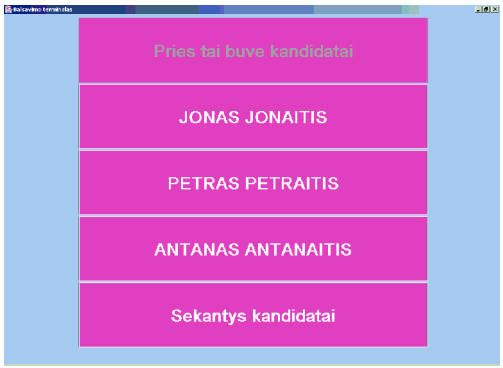
53 pav. Balsavęs rinkėjas



54 pav. Neprisiregistravęs rinkėjas



55 pav. Neegzistuojantis rinkėjas

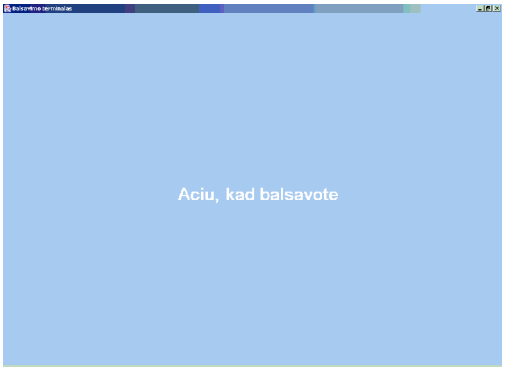


56 pav. Kandidato rinkimas

Kandidato rinkimo lange (56 pav.) vaizduojamas kandidatų sąrašas. Vienu metu vaizduojami trys kandidatai. Esant daugiau kandidatų nei trys teks spausti mygtukus „Sekantys kandidatai“ arba „ Pries tai buve kandidatai“. Jei kuris vienas iš šių dviejų mygtukų yra su papilkėjusiomis raidėmis, reiškia, kad kandidatų sąrašo ta linkme slinkti nebegalima. Paveiksle pavaizduotu atveju, slinkti negalima atgal, nes mygtuko „Pries tai buve kandidatai“ užrašas papilkėjęs. Pasirinkus norimą kandidatą atsiranda langas prašantis patvirtinti jūsų pasirinkimą (57 pav.).

57 pav. Pasirinkimo patvirtinimas

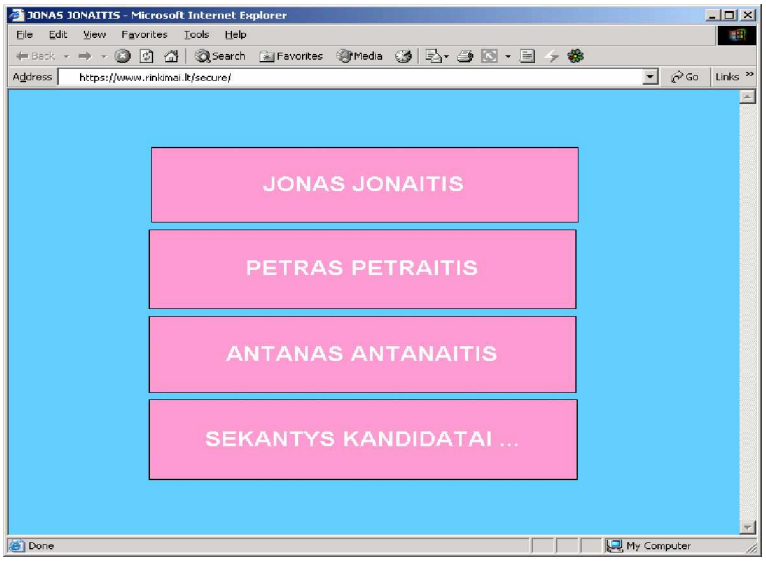
Atsakius į klausima „Ne“ grįžtama į ankstesnį langą, o jei taip tada 10 sekundžių dega langas pranešantis, kad pilietis atliko savo pilietinę pareigą (58 pav.) po to pereinama į pradinį balsavimo langą (47 pav.).



58 pav. Pilietinė pareiga atlikta

Piliečiams taip pat suteikiama galimybė balsuoti alternatyviais būdais: WAP sąsaja (59 pav.), internetu (60 pav.)., taip pat trumposiomis žinutėmis (SMS) (61 pav.).

59 pav. Balsavimas WAP sąsaja.



60 pav. Balsavimas internetu.

61 pav. Balsavimas trumposiomis žinutėmis (SMS).

4.2. Tolimesnio sistemos tobulinimo, plėtojimo galimybės

Mobiliųjų E-rinkimų taikomąją sistemą galima plėsti ir tobulinti. Galima integruoti naujus balsavimo būdus. Asmenų identifikavimui įmanoma panaudoti pirštų antspaudų arba akies rainelės nuskaitymą. Kadangi technologijos vystosi labai sparčiai, greitai visa tai galėsime patys išbandyti ir po kelių metų tai bus įprasti dalykai. Sistemą galima pritaikyti įvairiems rinkimų tipams, tiek prezidento rinkimams, tiek vidiniam balsavimui seime, tiek įvairioms žmonių apklausoms. Pagrindiniai pakeitimai turi būti padaryti programinėje dalyje. Techninė dalis gali būti naudojama nedarant didesnių investicijų ir atnaujinimų.

5. Išvados

Analizuojama tradicinių rinkimų situacija Lietuvoje ir apžvelgta jų silpnoji bei stiprioji pusės. Išanalizuotos E-rinkimų taikomosios sistemos įdiegimo galimybės bei pateikti jau realizuoti panašių sistemų pavyzdžiai su apibendrintais aprašymais. Analitinėje dalyje supažindinama su rinkimų organizavimo įstatymine baze.

Sumodeliuota konkreti rinkimų aplinka su duomenų ir procesų srautų diagramomis, pateikta išsami techninės bazės reikalingos realizuoti E-rinkimams specifikacija. Sukurti duomenų bazių modeliai.

Pasitelkiant Java™ programavimo kalbos priemones sukurta eksperimentinė bandomoji E-rinkimų taikomoji sistema. Realizuota tarpusavyje sujungta centrinės darbo (client-server technologija) stoties, apygardos darbo stoties ir apygardos darbo stoties hierarchinė struktūra. Sukurtas balsavimo terminalo modelis skirtas betarpiškam bendravimui su rinkėju.

Taigi, galima teigti, kad kompiuterinių rinkimų realizacija išties yra įmanoma naudojantis šiuo metu prieinamomis techninėmis ir programinėmis priemonėmis. Mūsų siūlomas modelis galėtų būti pavyzdinis sistemos variantas. Galime drąsiai sakyti, kad kompiuterizavę rinkimus ne tik pasieksime aukštą rezultatų tikslumą, bet ir užtikrinsime didelį rinkėjų aktyvumą. Tai pasieksime realizavę visiems prieinamą balsavimą bei sudominę žmones rinkimų naujovėmis.

6. Literatūra

[1] Lietuvos Respublikos Seimas. Elections to the Seimas. Iš *Lietuvos Respublikos Seimo Informacijos technologijų departamento* [interaktyvus]. 2004. [žiūrėta: 2004-11-20]. Prieiga per internetą: http://rc.lrs.lt/n/rinkimai/20001008/index_en.html

[2] Vyriausioji rinkimų komisija. Vyriausiosios rinkimų komisijos įgaliojimai. Iš *Vyriausioji rinkimų komisija* [interaktyvus]. 2004. [žiūrėta: 2004-11-20]. Prieiga per internetą: http://www.vrk.lt/vrk_igaliojimai.htm

[3] Sun Microsystems. The Source for Java Technology. Iš *Sun Microsystems* [interaktyvus]. 2003. [žiūrėta: 2004-11-25]. Prieiga per internetą: http://java.sun.com/

[4] Marc Fournier, Tom Lane, Vadim Mikheev. Postgresql. Iš *Postgresql* [interaktyvus]. 2003. [žiūrėta: 2004-11-25]. Prieiga per internetą: http://www.postgresql.org

[5] VoteHere. VoteHere. Iš *VoteHere* [interaktyvus]. 2004. [žiūrėta: 2004-09-15]. Prieiga per internetą: http://www.votehere.net

[6] Omnitel. Rinkimų svetainė Balsas. Iš *Omnitel* [interaktyvus]. 2005. [žiūrėta: 2004-11-30]. Prieiga per internetą: http://www.balsas.lt/

[7] BBC News. E-voting: A load of old ballots? Iš *BBC* [interaktyvus]. 2003. [žiūrėta: 2004-09-15]. Prieiga per internetą: http://news.bbc.co.uk/2/hi/in_depth/sci_tech/2000/dot_life/1746902.stm

[8] Rebecca Mercuri, Ph.D.. Electronic Voting. Iš *Notable Software* [interaktyvus]. 2004. [žiūrėta: 2004-09-16]. Prieiga per internetą: http://www.notablesoftware.com/evote.html

[9] Lorrie Cranor. Electronic Voting Hot List. Iš *Electronic Voting Hot List* [interaktyvus]. 2004. [žiūrėta: 2004-09-16]. Prieiga per internetą: http://lorrie.cranor.org/voting/hotlist.html

[10] David Chaum. Sure Vote. Iš *Sure Vote* [interaktyvus]. 2004. [žiūrėta: 2004-09-17]. Prieiga per internetą: http://www.sure-vote.com/home.html

[11] David Chaum. SafeVote. Iš *SafeVote* [interaktyvus]. 2004. [žiūrėta: 2004-09-17]. Prieiga per internetą: http://www.safevote.com/

[12] ES&S. Election systems and software. Iš *ES&S* [interaktyvus]. 2004. [žiūrėta: 2004-09-25]. Prieiga per internetą: http://www.essvote.com/

[13] ES&S. Election systems and software. Iš *ES&S* [interaktyvus]. 2004. [žiūrėta: 2004-09-25]. Prieiga per internetą: http://www.votingsolutions.com/

[14] MTI. Technology for the next Millennium. Iš *MTI* [interaktyvus]. 2004. [žiūrėta: 2004-09-25]. Prieiga per internetą: http://www.votingsystems.com/

[15] The global election company. Iš *The global election company* [interaktyvus]. 2004. [žiūrėta: 2004-09-30]. Prieiga per internetą: http://www.election.com/uk/index.htm

[16] Reply. Reply. Iš *ReplySystems* [interaktyvus]. 2004. [žiūrėta: 2004-09-30]. Prieiga per internetą: http://www.replysystems.com/

[17] MicroVote general corporation. Iš *MicroVote general corporation* [interaktyvus]. 2004. [žiūrėta: 2004-09-30]. Prieiga per internetą: http://www.microvote.com/

7. Terminų ir santrumpų žodynas

UML („Unified Modeling Language“) – speciali kalba skirta programinių sistemų, biznio modelių ir kitų ne programinių sistemų specifikavimui, vizualizavimui, konstravimui ir dokumentavimui.

RSA – tai viešojo rakto šifravimo sistema skirta šifravimui ir autentifikavimui.

Real-time – tai ka vyksta nedelsiant. Terminas naudojamas apibrėžti specifinėms kompiuterio funkcijoms, tokioms kaip realaus laiko operacinės sistemos ir panašiai.

Applet – tai Java programavimo kalba parašyta programa, kuri vykdoma internetinėje naršyklėje ar vykdoma Java interpretavimo programos.

Servlet – tai Java programavimo kalba parašyta programa vykdoma specialioje Java Internetinio Serverio aplinkoje.

Proxy – darbinė stotis, esanti tarp klientinės programos, pvz. internetinė naršyklė ir realaus serverio. Stotis perima užklausas ir tikrina ar pati turimais resursais gali jas įvykdyti.

URL („Uniform Resource Location“) – globalus adresas iki konkretaus dokumento ar kito resurso talpinamo globaliame internetiniame tinkle.

TCP („Transmission Control Protocol“) – tai pagrindinis protokolas TCP/IP tinkluose. Tai protokolas, užtikrinantis sujungimą tarp dviejų aktyvių tinklo elementų. Protokolas užtikrina duomenų pilnumą ir sekos eiliškumą.

UDP („User Datagram Protocol“) – TCP/IP tinklo ryšio protokolas. Užtikrina žemą klaidų atstatymą siunčiant tiesioginius duomenų paketus tinkle. Daugiausia naudojamas garsinei informacijai tinkle perduoti.

Socket – programinis objektas prijungiantis programą prie tinklo protokolų. Duomenų mainai vyksta skaitant ir rašant informaciją į šį objektą.

IP („Internet Protocol“) – tinklo protokolas, apibrėžiantis perduodamus informacijos paketus, nukreipimo taisykles bei šaltinius/

JavaBeans™ – kompanijos Sun Microsystems sukurtas formatas (specifikacija), aprašantis Java kalba aprašytų objektų sąveikas.

VPN („Virtual Private Network“) – technologija skirta tinklo elementams apjungti į vientisą saugų (šifruojamą) tinklą. Naudojami specialūs kodavimo ir saugumo mechanizmai tam, kad užtikrinti vartotojų autentiškumą bei duomenų saugumą.

HTTPS („Secure HyperText Transfer Protocol“) – tinklo protokolas, užtikrinantis saugų duomenų perdavimą internetu.

SMS („Short Message Service“) – trumpųjų žinučių paslauga pranešimų persiuntimui mobiliaisiais tinklais.

WAP („Wireless Application Protokol“) – protokolas, leidžiantis mobiliųjų įrenginių vartotojams priimti informaciją iš internetinio tinklo. Naudojamas mobiliuosiuose telefonuose, delniniuose kompiuteriuose, komunikatoriuose ir kitur.

Ugniasienė – sistema, skirta apsaugoti priėjimą prie sistemos resursų nuo neautorizuotų sistemų. Gali būti realizuotas tiek techninėmis, tiek ir programinėmis priemonėmis.

Smart Card – mažas elektroninis įrenginys (dažniausiai kreditinės kortelės dydžio), turintis atminties bei papildomos integruotos logikos elementus

WLAN („Wireless Local Area Network“) – lokalaus tinklo tipas, vietoj specialių kabelių, naudojantis aukšto dažnio radijo bangas ryšiui tarp tinklo taškų.
Maršrutizatorių – įrenginys, persiunčiantis duomenų paketus tinkle. Skirtos keletos kompiuterinių tinklų apjungimui.

Bar Code – specialių skaitytuvų atpažįstamas universalus kodas, naudojamas objekto atpažinimui.

8. Santrauka anglų kalba

Mobile E-voting application system - software and hardware solution for e-voting elections. We have reviewed and analyzed present electronic voting systems, that were realized on public elections. There are many e-voting systems, but no one of them reach today's needs, because all of them has just one, two or three voting ways, but no one has almost all and most needed of them. Another problem is security. We have created two levels security solution, to secure votes in the e-voting terminals, until they will be send to then central voting system. All mobile e-voting application system has three levels, then first is central voting station, the second is district voting station and the last is terminal. The data from the central station to the district station goes through the VPN (Virtual Private Network) and from the district station to the terminal goes through the WLAN modem. We have formed technical, information, user interface requirements and models for the mobile E-voting application system. There are described possible solutions for computer network, database and user interface. Experimental example shows a few Mobile E-voting application system functions.

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Audrius Leipus
Arvydas Vapsva

Mobiliųjų E-rinkimų taikomoji sistema

Priedai

Darbo vadovas
doc. V. Kiauleikis

Kaunas
2005

Priedas Nr. 1

Analysis of an Electronic Voting System

TADAYOSHI KOHNO_ ADAM STUBBLEFIELD† AVIEL D. RUBIN‡
DAN S. WALLACH§

February 27, 2004

Abstract

With significant U.S. federal funds now available to replace outdated punch-card and mechanical voting systems, municipalities and states throughout the U.S. are adopting paperless electronic voting systems from a number of different vendors. We present a security analysis of the source code to one such machine used in a significant share of the market. Our analysis shows that this voting system is far below even the most minimal security standards applicable in other contexts. We identify several problems including unauthorized privilege escalation, incorrect use of cryptography, vulnerabilities to network threats, and poor software development processes. We show that voters, without any insider privileges, can cast unlimited votes without being detected by any mechanisms within the voting terminal software. Furthermore, we show that even the most serious of our outsider attacks could have been discovered and executed without access to the source code. In the face of such attacks, the usual worries about insider threats are not the only concerns; outsiders can do the damage. That said, we demonstrate that the insider threat is also quite considerable, showing that not only can an insider, such as a poll worker, modify the votes, but that insiders can also violate voter privacy and match votes with the voters who cast them. We conclude that this voting system is unsuitable for use in a general election. Any paperless electronic voting system might suffer similar flaws, despite any "certification" it could have otherwise received. We suggest that the best solutions are voting systems having a "voter-verifiable audit trail," where a computerized voting system might print a paper ballot that can be read and verified by the voter.

_Dept. of Computer Science and Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California
92093, USA. E-mail: tkohno@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/tkohno. Most of this work was performed while visiting the Johns Hopkins University Information Security Institute. Supported by a National Defense Science
and Engineering Graduate Fellowship.

†Information Security Institute, Johns Hopkins University, 3400 North Charles Street, Baltimore, Maryland 21218, USA. Email:
astubble@cs.jhu.edu. URL: http://spar.isi.jhu.edu/˜astubble.

‡Information Security Institute, Johns Hopkins University, 3400 North Charles Street, Baltimore, Maryland 21218, USA. Email:
rubin@cs.jhu.edu. URL: http://www.avirubin.com.

§Dept. of Computer Science, Rice University, 3121 Duncan Hall, 6100 Main Street, Houston, Texas 77005, USA. E-mail:
dwallach@cs.rice.edu. URL: http://www.cs.rice.edu/˜dwallach.

Contents

1 Introduction 3

2 System overview 5

3 Smartcards 9

3.1 Exploiting the lack of cryptography: Creating homebrew smartcards 9

3.2 Casting multiple votes 10

3.3 Accessing administrator and poll worker functionality 10

4 Election configurations and election data 11
4.1 Tampering with the system configuration 12
4.2 Tampering with ballot definitions 13
4.3 Impersonating legitimate voting terminals 14
4.4 Key management and other cryptographic issues with the vote and audit records 14
4.5 Tampering with election results and linking voters with their votes 15
4.6 Audit logs 17
4.7 Attacking the start of an election 17
5 Software engineering 18
5.1 Code legacy 18
5.2 Coding style 18
5.3 Coding process 19
5.4 Code completeness and correctness 20
6 Conclusions 21
2

1 Introduction

Elections allow the populace to choose their representatives and express their preferences for how they will be governed. Naturally, the integrity of the election process is fundamental to the integrity of democracy itself. The election system must be sufficiently robust to withstand a variety of fraudulent behaviors and must be sufficiently transparent and comprehensible that voters and candidates can accept the results of an election. Unsurprisingly, history is littered with examples of elections being manipulated in order to influence their outcome.

The design of a "good" voting system, whether electronic or using traditional paper ballots or mechanical devices, must satisfy a number of sometimes competing criteria. The *anonymity* of a voter's ballot must be preserved, both to guarantee the voter's safety when voting against a malevolent candidate, and to guarantee that voters have no evidence that proves which candidates received their votes. The existence of such evidence would allow votes to be purchased by a candidate. The voting system must also be *tamper-resistant* to thwart a wide range of attacks, including ballot stuffing by voters and incorrect tallying by insiders. Another factor, as shown by the so-called "butterfly ballots" in the Florida 2000 presidential election, is the importance of *human factors*. A voting system must be comprehensible to and usable by the *entire* voting population, regardless of age, infirmity, or disability. Providing accessibility to such a diverse population is an important engineering problem and one where, if other security is done well, electronic voting could be a great improvement over current paper systems. Flaws in any of these aspects of a voting system, however, can lead to indecisive or incorrect election results.

ELECTRONIC VOTING SYSTEMS. There have been several studies on using computer technologies to improve
elections [4, 5, 20, 21, 25]. These studies caution against the risks of moving too quickly to adopt electronic voting machines because of the software engineering challenges, insider threats, network vulnerabilities,
and the challenges of auditing.

As a result of the Florida 2000 presidential election, the inadequacies of widely-used punch card voting systems have become well understood by the general population. Despite the opposition of computer scientists, this has led to increasingly widespread adoption of "direct recording electronic" (DRE) voting systems. DRE systems, generally speaking, completely eliminate paper ballots from the voting process. As with traditional elections, voters go to their home precinct and prove that they are allowed to vote there, perhaps by presenting an ID card, although some states allow voters to cast votes without any identification at all. After this, the voter is typically given a PIN, a smartcard, or some other token that allows them to approach a voting terminal, enter the token, and then vote for their candidates of choice. When the voter's

selection is complete, DRE systems will typically present a summary of the voter's selections, giving them a final chance to make changes. Subsequent to this, the ballot is "cast" and the voter is free to leave.

The most fundamental problem with such a voting system is that the entire election hinges on the correctness, robustness, and security of the software within the voting terminal. Should that code have securityrelevant flaws, they might be exploitable either by unscrupulous voters or by malicious insiders. Such insiders include election officials, the developers of the voting system, and the developers of the embedded operating system on which the voting system runs. If any party introduces flaws into the voting system software or takes advantage of pre-existing flaws, then the results of the election cannot be assured to accurately reflect the votes legally cast by the voters.

Although there has been cryptographic research on electronic voting [13], and there are new approaches such as [6], currently the most viable solution for securing electronic voting machines is to introduce a "voter-verifiable audit trail" [10, 20]. A DRE system with a printer attachment, or even a traditional optical scan system (e.g., one where a voter fills in a printed bubble next to their chosen candidates), will satisfy this requirement by having a piece of paper for voters to read and verify that their intent is correct reflected. This paper is stored in ballot boxes and is considered to be the primary record of a voter's intent. If, for

some reason, the printed paper has some kind of error, it is considered to be a "spoiled ballot" and can be mechanically destroyed, giving the voter the chance to vote again. As a result, the correctness of any voting software no longer matters; either a voting terminal prints correct ballots or it is taken out of service. If there is any discrepancy in the vote tally, the paper ballots will be available to be recounted, either mechanically or by hand. (A verifiable audit trail does not, by itself, address voter privacy concerns, ballot stuffing, or numerous other attacks on elections.)

"CERTIFIED" DRE SYSTEMS. Many government entities have adopted paperless DRE systems without appearing to have critically questioned the security claims made by the systems' vendors. Until recently, such systems have been dubiously "certified" for use without any public release of the analyses behind these certifications, much less any release of the source code that might allow independent third parties to perform their own analyses. Some vendors have claimed "security through obscurity" as a defense, despite the security community's universally held belief in the inadequacy of obscurity to provide meaningful protection [18].

Indeed, the CVS source code repository for Diebold's AccuVote-TS DRE voting system recently appeared on the Internet. This appearance, announced by Bev Harris and discussed in her book, *Black Box Voting* [14], gives us a unique opportunity to analyze a widely used, paperless DRE system and evaluate the manufacturer's security claims. Jones discusses the origins of this code in extensive detail [17]. Diebold's voting systems are in use in 37 states, and they are the second largest and the fastest growing vendor of electronic voting machines. We only inspected unencrypted source code, focusing on the AVTSCE, or AccuVote-TS version 4, tree in the CVS repository [9]. This tree has entries dating from October 2000 and culminates in an April 2002 snapshot of version 4.3.1 of the AccuVote-TS system. From the comments in the CVS logs, the AccuVote-TS version 4 tree is an import of an earlier AccuTouch-CE tree. We did not have source code to Diebold's GEMS back-end election management system.

SUMMARY OF RESULTS. We discovered significant and wide-reaching security vulnerabilities in the version of the AccuVote-TS voting terminal found in [9] (see Table 1). Most notably, voters can easily program their own smartcards to simulate the behavior of valid smartcards used in the election. With such homebrew cards, a voter can cast multiple ballots without leaving any trace. A voter can also perform actions that normally require administrative privileges, including viewing partial results and terminating the election early. Similar undesirable modifications could be made by malevolent poll workers (or janitorial staff) with access to the voting terminals before the start of an election. Furthermore, the protocols used when the voting terminals

communicate with their home base, both to fetch election configuration information and to report final election results, do not use cryptographic techniques to authenticate either end of the connection nor do they check the integrity of the data in transit. Given that these voting terminals could potentially communicate over insecure phone lines or even wireless Internet connections, even unsophisticated attackers can perform untraceable "man-in-the-middle" attacks.

We considered both the specific ways that the code uses cryptographic techniques and the general software engineering quality of its construction. Neither provides us with any confidence of the system's correctness. Cryptography, when used at all, is used incorrectly. In many places where cryptography would seem obvious and necessary, none is used. More generally, we see no evidence of disciplined software engineering processes. Comments in the code and the revision change logs indicate the engineers were aware of some areas in the system that needed improvement, though these comments only address specific problems with the code and not with the design itself. We also saw no evidence of any change-control process that might restrict a developer's ability to insert arbitrary patches to the code. Absent such processes, a malevolent developer could easily make changes to the code that would create vulnerabilities to be later exploited on Election Day. We also note that the software is written entirely in C++. When programming in a language like C++, which is not type-safe, programmers must exercise tight discipline to prevent their programs from being vulnerable to buffer overflow attacks and other weaknesses. Indeed, buffer overflows

Figure 1: A Diebold AccuVote-TS voting machine (photo from http://www.sos.state.ga.us/). Note the smartcard reader in the lower-right hand corner.

caused real problems for AccuVote-TS systems in real elections.[1]

SUBSEQUENT WORK. Following the release of our results, the state of Maryland hired SAIC [27] and RABA [24] and the state of Ohio hired Compuware [7] to perform independent analyses of Diebold's AccuVote-TS systems. These analyses not only support our findings, but show that many of the issues we raise and attacks we identify still apply to recent versions of the AccuVote-TS system, and particularly to the machines recently purchased by Maryland. These analyses also identified security problems with the back-end GEMS server. Additionally, RABA's "red team" implemented some of our attacks in a mock election setting; e.g., they modified smartcards so that a voter could vote more than once (Section 3.2 and [24, page 16]) and they implemented our ballot reordering attack, thereby tricking voters to vote for the wrong candidates (Section 4.2 and [24, pages 18 and 21]). Jones discusses these three reports in more detail [17].

2 System overview

The Diebold AccuVote-TS 4.3.1 system we analyzed [9], which was written in C++, was designed to run on a Windows CE device, an example of which is shown in Figure 1. The code also compiles and runs (with slightly different configurations) on regular Microsoft Windows machines, thus enabling us to verify that the code represents a complete system. We shall refer to a device running the vote collection software as a *voting terminal*.

1http://www.sccgov.org/scc/assets/docs/209815KeyboardAttachment-200440211.pdf (page 60 of the report, page 61 of the PDF)

Voter (with forged smartcard)	Poll Worker (with access to storage media)	Poll Worker (with access to network traffic)	Internet Provider (with access to network traffic)	OS Developer	Voting Device Developer	Section
Vote multiple times	•	•	•			3.2
using forged smartcard						
Access administrative functions	•	•	•	•		3.3
or close polling station						
Modify system configuration	•	•				4.1

Modify ballot definition	• • • • • 4.2
(e.g., party affiliation)	
Cause votes to be miscounted	• • • • • 4.2
by tampering with configuration	
Impersonate legitimate voting	• • • • • 4.3
machine to tallying authority	
Create, delete, and modify votes	• • • • • 4.3, 4.5
Link voters with their votes	• • • • • 4.5
Tamper with audit logs	• • • 4.6
Delay the start of an election	• • • • • 4.7
Insert backdoors into code	• • 5.3

Table 1: This table summarizes some of the more important attacks on the system.

Below we describe the process for setting up and running an election using the Diebold system. In some cases, where election procedures and policies might vary or where we have insufficient information from studying the code, we will state our assumptions. We note that, even in cases where election policies and procedures might provide protection against design shortcomings, those policies and procedures depend on poll workers who may not fully understand or be able to carry out their responsibilities. As a result, any failure in the design of the voting system may very well be abused to compromise an election.

SETTING UP. Before an election takes place, one of the first things the election officials must do is specify the political offices and issues to be decided by the voters along with the candidates and their party affiliations. Variations on the ballot can be presented to voters based on their party affiliations. We call this data a *ballot definition*. In the Diebold system, a ballot definition is encoded as the file election.edb.

Prior to an election, the voting terminals must be configured and installed at each voting location. A governmental entity using Diebold voting terminals has a variety of choices in how to distribute the ballot definitions. They also may be distributed using removable media, such as floppy disks or storage cards, or over a local network, the Internet, or a dial-up connection. The networked approach, if allowed under the voting precinct's processes, provides additional flexibility to the election administrator in the event of last-minute changes to the ballot.

THE ELECTION. Once the voting terminal is initialized with the ballot definitions and the election begins, voters are allowed to cast their votes. To get started, the voter must have a *voter card*. The voter card is a memory card or smartcard; i.e., it is a credit-card sized plastic card with a computer chip on it that can store data and, in the case of the smartcard, perform computation. Under the most common scenario, we assume that the voting cards are given to voters at the voting site on election day.

The voter takes the voter card and inserts it into a smartcard reader attached to the voting terminal. The terminal checks that the smartcard in its reader is a voter card and, if it is, presents a ballot to the voter on the terminal screen. The actual ballot the voter sees may depend on the voter's political party, which is encoded on the voter card. If a ballot cannot be found for the voter's party, the voter is given a nonpartisan ballot. Such party-specific ballots are used, for example, in primaries.

At this point, the voter interacts with the voting terminal, touching the appropriate boxes on the screen for his or her desired candidates. Headphones and keypads are available for visually-impaired voters to privately interact with the terminal. Before the ballots are committed to storage in the terminal, the voter is given a final chance to review his or her selections. If the voter confirms this, the vote is recorded on the voting terminal and the voter card is "canceled." This latter step is intended to prevent the voter from voting again with the same card. After the voter finishes voting, the terminal is ready for another voter to use. The voter returns his or her canceled card to the poll workers, who reprogram it for the next user.

REPORTING THE RESULTS. A poll worker ends the election process by inserting an *administrator card*

or an *ender card* (a special card that can only be used to end the election) into the voting terminal. Upon detecting the presence of such a card (and, in the case of the administrator card, checking a PIN entered by the card user), the poll worker is asked to confirm that the election is finished. If the poll worker agrees, then the voting terminal enters the post-election stage. Election results are written to a removable flash memory card and can also be transmitted electronically to the back-end server.

As we have only analyzed the code for the Diebold voting terminal, we do not know exactly how the back-end server tabulates the final results it gathers from the individual terminals. Obviously, it collects all the votes from the various voting terminals. We are unable to verify that there are checks to ensure, for example, that there are no more votes collected than people who are registered at or have entered any given polling location.

DETAILED OVERVIEW OF THE CODE. The 4.3.1 snapshot of the AccuVote-TS tree [9] has 136 .h files totaling 16414 lines and 120 .cpp files totaling 33195 lines, for a total of 256 files and 49609 lines of C++

code. While a full description of every module in the Diebold AccuVote-TS 4.3.1 system is beyond the scope of this paper, we describe the bootstrapping process as well as the main state transitions that occur within a Diebold system during an election, making explicit references to the relevant portions of the code. The voting terminal is implemented in the directory BallotStation/, but uses libraries in the supporting directories Ballot/, DES/, DiagMode/, Shared/, TSElection/, Utilities/, and VoterCard/.

The method CBallotStationApp::DoRun() is the main loop for the voting terminal software. The DoRun() method begins by invoking CBallotStationApp::LoadRegistry(), which loads information about the voting terminal from the registry (the registry keys are stored under HKEY_LOCAL_MACHINE\Software\Global Election Systems\AccuVote-TS4) . If the program fails to load the registry information, it believes that it is uninitialized and therefore creates a new instance of the CTSRegistryDlg class that asks the administrator to set up the machine for the first time. The administrator chooses, among other things, the COM port to use with the smartcard reader, the directory locations to store files, and the polling location identifier. The CBallotStationApp::DoRun() method then checks for the presence of a smartcard reader and, if none is found, gives the administrator the option to interact with the CTSRegistryDlg again.

The DoRun() method then enters a while loop that iterates until the software is shut down. The first thing DoRun() does in this loop is check for the presence of some removable media on which to store election results and ballot configurations (a floppy under Windows or a removable storage card on a Windows CE device). It then tries to open the election configuration file election.edb. If it fails to open the configuration file, the program enters the CTSElectionDoc::ES_NOELECTION state and invokes CBallotStationApp::Download(), which creates an instance of CTransferElecDlg to download the configuration file. To do the download, the terminal connects to a back-end server using either the Internet or a dial-up connection. Subsequently, the program enters the CTSElectionDoc::ES_PREELECT state, invoking the CBallotStationApp::PreElect() method, which in turn creates an instance of CPreElectDlg. The administrator can then decide to start the election, in which case CPreElectDlg::OnSetForElection() sets the state of the terminal to CTSElectionDoc::ES_ELECTION.

Returning to the while loop in CBallotStationApp::DoRun(), now that the machine is in the state CTSElectionDoc::ES_ELECTION, the DoRun() method invokes CBallotStationApp::Election(), which creates an instance of CVoteDlg. When a card is inserted into the reader, the application checks to see if the card is a voter card, administrator card, or ender card. If it is an ender card, or if it is an administrator card and if the user enters the correct PIN, the CVoteDlg ends and the user is asked whether he or she wishes to terminate the election and, if so, the state of the terminal is set to CTSElectionDoc::ES_POSTELECT. If the user entered a voter card, then DoVote() is invoked

(here DoVote() is an actual function; it does not belong to any class). The DoVote() function finds the appropriate ballot for the user's voter group or, if none exists, opens the nonpartisan ballot (recall that the system is designed to support different ballots for different voters, as might occur in a primary party election). It then creates an instance of CBallotDlg to display the ballot and collect the votes.

We recall that if, during the election process, someone inserted an administrator or ender card into the terminal and chooses to end the election, the system would enter the CTSElectionDoc::ES_POSTELECT state. At this point the voting terminal would offer the ability to upload the election results to some back-end server for final tabulation. The actual transfer of results is handled by the CTransferResultsDlg::OnTransfer() method.

3 Smartcards

While it is true that one can design secure systems around the use of smartcards, merely the use of smartcards in a system does *not* imply that the system is secure. The system must use the smartcards in an intelligent and security-conscious way. Unfortunately, the Diebold system's use of smartcards provides very little (if any) additional security and, in fact, opens the system to several attacks.

3.1 Exploiting the lack of cryptography: Creating homebrew smartcards

Upon reviewing the Diebold code, we observed that the smartcards do not perform any cryptographic operations.

This, in and of itself, is an immediate red flag. One of the biggest advantages of smartcards over classic magnetic-stripe cards is the smartcards' ability to perform cryptographic operations internally, and with physically protected keys. Because of a lack of cryptography, *there is no secure authentication of the smartcard to the voting terminal.* This means that nothing prevents an attacker from using his or her own homebrew smartcard in a voting terminal. One might naturally wonder how easy it would be for an attacker to make such a homebrew smartcard. First, we note that user-programmable smartcards and smartcard readers are available commercially over the Internet in small quantities and at reasonable prices. Second, an attacker who knows the protocol spoken between voting terminals and legitimate smartcards could easily implement a homebrew card that speaks the same protocol. We shall shortly consider how an attacker might go about learning the protocol if he or she does not know it *a priori*.

Once the adversary knows the protocol between the terminal and the smartcards, the only impediment to the mass production of homebrew smartcards is that each voting terminal will make sure that the smartcard has encoded in it the correct m_ElectionKey, m_VCenter, and m_DLVersion (see DoVote() in BallotStation/Vote.cpp). The m_ElectionKey and m_DLVersion are likely the same for all locations and, furthermore, for backward-compatibility purposes it is possible to use a card with m_ElectionKey and m_DLVersion undefined. The m_VCenter value could be learned on a perlocation-basis by interacting with legitimate smartcards, from an insider, or from inferences based on the m_VCenter values observed at other polling locations. In short, all the necessary information to create homebrew counterfeit smartcards is readily available.

In the next subsections we consider attacks that an adversary could mount after creating homebrew cards. We find the issues we uncovered to be particularly distressing as modern smartcard designs allow cryptographic operations to be performed directly on the smartcard, making it possible to create systems that are not as easily vulnerable to such security breaches.

REVERSE ENGINEERING THE SMARTCARD PROTOCOL. It turns out that adversaries, including regular voters, who do not know *a priori* the protocol between the smartcard and the terminal can "easily" learn the protocol, thereby allowing them to produce homebrew voter cards. An adversary, such as a poll worker, with the ability to interact with a legitimate administrator or ender card could also learn enough information to produce homebrew administrator and ender cards (Section 3.3).

Let us consider several ways that an adversary could learn the protocol between voter cards and voting terminals. After voting, instead of returning the canceled card to the poll-worker, the adversary could return a fake card that records how it is reprogrammed, and then dumps that information to a collaborating attacker waiting in line to vote. Alternatively, the attacker could attach a "wiretap" device between the voting terminal and a legitimate smartcard and observe the communicated messages. The parts for building such a device are readily available and, depending on the setup at each voting location, might be unnoticed by poll workers. An attacker might not even need to use a wiretap device: as a literal "person-in-the-middle" attack, the adversary could begin by inserting a smartcard into the terminal that records the terminal's first message. The adversary would then leave the voting location, send that message to a real voter card that he or she stole, and learn the real voter card's response. The adversary's conspirator could then show up at the

voting location and use the information gained in the first phase to learn the next round of messages in the protocol, and so on. We comment again that these techniques work because the authentication process is completely deterministic and lacks any sort of cryptography.

3.2 Casting multiple votes

In the Diebold system, a voter begins the voting process by inserting a smartcard into the voting terminal. Upon checking that the card is "active," the voting terminal collects the user's vote and then deactivates the user's card; the deactivation actually occurs by rewriting the card's type, which is stored as an 8-bit value on the card, from VOTER_CARD (0x01) to CANCELED_CARD (0x08). Since an adversary can make perfectly valid smartcards, the adversary could bring a stack of active cards to the voting booth. Doing so gives the adversary the ability to vote multiple times. More simply, instead of bringing multiple cards to the voting booth, the adversary could program a smartcard to ignore the voting terminal's deactivation command. Such an adversary could use one card to vote multiple times. Note here that the adversary could be a regular voter, and not necessarily an election insider.

Will the adversary's multiple-votes be detected by the voting system? To answer this question, we must first consider what information is encoded on the voter cards on a per-voter basis. The only pervoter information is a "voter serial number" (m_VoterSN in the CVoterInfo class). m_VoterSN is only recorded by the voting terminal if the voter decides *not* to place a vote (as noted in the comments in TSElection/Results.cpp, this field is recorded for uncounted votes for backward compatibility reasons). It is important to note that if a voter decides to cancel his or her vote, the voter will have the opportunity to vote again using that same card (and, after the vote has been cast, m_VoterSN will no longer be recorded).

If we assume the number of collected votes becomes greater than the number of people who showed up to vote, and if the polling locations keep accurate counts of the number of people who show up to vote, then the back-end system, if designed properly, should be able to detect the existence of counterfeit votes. However, because m_VoterSN is only stored for those who did not vote, there will be no way for the tabulating system to distinguish the real votes from the counterfeit votes. This would cast serious doubt on the validity of the election results. The solution proposed by one election official, to have everyone vote again, does not seem like a viable solution.

3.3 Accessing administrator and poll worker functionality

As noted in Section 2, in addition to the voter cards that normal voters use when they vote, there are also administrator cards and ender cards, which have special purposes in this system. The administrator cards give the possessor the ability to access administrative functionality (the administrative dialog BallotStation/AdminDlg.cpp), and both types of cards allow the possessor to end the election (hence the term "ender card").

Just as an adversary can manufacture his or her own voter cards, an adversary can manufacture his or her own administrator and ender cards (administrator cards have an easily-circumventable PIN, which we will discuss shortly). This attack is easiest if the attacker has knowledge of the Diebold code or can interact with

a legitimate administrator or ender card, since otherwise the attacker would not know what distinguishes an administrator or ender card from a voter card. (The distinction is that, for a voter card m_CardType is set to 0x01, for an ender card the value is 0x02, and for an administrator card the value is 0x04.)

As one might expect, an adversary in possession of such illicit cards has further attack options against the Diebold system. Using a homebrew administrator card, a poll worker, who might not otherwise have access to the administrator functions of the Diebold system but who does have access to the voting machines before and after the elections, could gain access to the administrator controls. If a malicious voter entered an

10

administrator or ender card into the voting device instead of the normal voter card, then the voter would be able to terminate the election and, if the card is an administrator card, gain access to additional administrative controls.

The use of administrator or ender cards prior to the completion of the actual election represents an interesting denial-of-service attack. Once "ended," the voting terminal will no longer accept new voters (see CVoteDlg::OnCardIn()) until the terminal is somehow reset. Such an attack, if mounted simultaneously by multiple people, could temporarily shut down a polling place. If a polling place is in a precinct considered to favor one candidate over another, attacking that specific polling place could benefit the lessfavored

candidate. Even if the poll workers were later able to resurrect the systems, the attack might succeed in deterring a large number of potential voters from voting (e.g., if the attack was performed over the lunch hour). If such an attack was mounted, one might think the attackers would be identified and caught. We note that many governmental entities, e.g., California, do not require identification to be presented by voters. By the time the poll workers realize that one of their voting terminals has been disabled, the perpetrator may have long-since left the scene. Furthermore, the poll workers may not be computer savvy and might simply think that all the machines crashed simultaneously.

CIRCUMVENTING THE ADMINISTRATOR PIN. In order to use (or create) an administrator card, the attacker

must know the PIN associated (or to be associated) with the card. Because the system's use of smartcards was poorly designed, an adversary could easily learn the necessary information, thereby circumventing any security the PIN might have offered.

We first note that the PIN is sent from the smartcard to the terminal in cleartext. As a result, anyone who knows the protocol and wishes to make their own administrator card could use any PIN of their choice. Even if the attacker does not know the protocol but has access to an existing administrator card and wants to make a copy, the adversary could guess the PIN in just a few trials if the adversary realizes that the PIN is included as part of a short cleartext message sent from the card. More specifically, rather than try all 10000 possibilities for the PIN, the adversary could try all 4-byte consecutive substrings of the cleartext message.

4 Election configurations and election data

In election systems, protecting the integrity and privacy of critical data (e.g., votes, configurations, ballot definitions) is undeniably important. We investigated how the Diebold system manipulates such data, and found considerable problems. There are two main vectors for accessing and attacking the voting system's data: via physical access to the device storing the data, or via man-in-the-middle attacks as the data is transported over some network. The latter assumes that the systems are connected to a network, which is possible though may be precluded by election procedures in some jurisdictions. Attacks via physical access to memory can be quite powerful, and can be mounted easily by insiders. The network attacks, which can also be quite powerful, can also be mounted by insiders as well as sophisticated outsiders.

DATA STORAGE OVERVIEW. Each voting terminal has two distinct types of internal data storage. A main (or system) storage area contains the terminal's operating system, program executables, static data files such as fonts, and system configuration information, as well as backup copies of dynamic data files such as the voting records and audit logs. Each terminal also contains a removable flash memory storage device that is

used to store the primary copies of these dynamic data files. When the terminal is running a standard copy of Windows (e.g., Windows 2000) the removable storage area is the first floppy drive; when the terminal is running Windows CE, the removable storage area is a removable storage card. Storing the dynamic data on two distinct devices is advantageous for both reliability and non-malleability: if either of the two storage mediums fails, data can still be recovered from the copy, although reconciling differences between these media may be difficult.

Unfortunately, in Windows CE, the existence of the removable storage device is not enforced properly.[11] Unlike other versions of Windows, removable storage cards are mounted as subdirectories under CE. When the voting software wants to know if a storage card is inserted, it simply checks to see if the Storage Card subdirectory exists in the filesystem's root directory. While this is the default name for a mounted storage device, it is also a perfectly legitimate directory name for a directory in the main storage area. Thus, if such a directory exists, the terminal can be fooled into using the same storage device for all of the data.[2] This would reduce the amount of redundancy in the voting system and would increase the chances that a hardware failure could cause recorded votes to be lost.

NETWORK OVERVIEW. The Diebold voting machines cannot work in isolation. They must be able to both receive a ballot definition file as input and report voting results as output. As described in Section 2, there are essentially two ways to load a voting terminal with an initial election configuration: via some removable media, such as a flash memory card, or over a network connection. In the latter case, the voting terminal could either be plugged directly into the Internet, could be connected to an isolated local network, or could use a dialup connection (the dial-up connection could be to a local ISP, or directly to the election authority's modem banks). Diebold apparently gives their customers a variety of configuration options; electronic networks are not necessary for the operation of the system. After the election is over, election results can be sent to a back-end post-processing server over the network (again, possibly through a dialup connection). When results are reported this way, it is not clear whether these network-reported results become the official results, or just the preliminary results (the official results being computed after the memory cards are removed from all the voting terminals and collected and tabulated at a central location).

We also observe that, even in jurisdictions where voting terminals are *never* connected to a network or phone line, the physical transportation of the flash memory cards from the voting terminal to the central tabulating system is really just a "sneaker net." Such physical card transportation must be robust against real-world analogies of network man-in-the-middle attacks. Any flaws in the policies and procedures used to protect the chain of custody could lead to opportunities for these cards to be read or written by an adversary. Consequently, even if no electronic computer network is used, we still view network attacks as critical in the design of a voting system.

4.1 Tampering with the system configuration

The majority of the system configuration information for each terminal is stored in the Windows registry under HKEY_LOCAL_MACHINE\Software\Global Election Systems\AccuVote-TS4 . This includes both identification information such as the terminal's serial number and more traditional configuration information such as the COM port to which the smartcard reader is attached. All of the configuration information is stored in the clear, without any form of integrity protection. Thus, all an adversary must do is modify the system registry to trick a given voting terminal into effectively impersonating any other voting terminal. It is unclear how the tabulating authority would deal with results from two different voting terminals with the same voting ID; at the very least, human intervention to resolve the conflict would probably be required.

The Federal Election Commission draft standard [11] requires each terminal to keep track of the total number of votes that have ever been cast on it — the "Protective Counter." This counter is used to provide yet another method for ensuring that the number of votes cast on each terminal is correct. However, as the

following code from Utilities/machine.cpp shows, the counter is simply stored as an integer in the file system.bin in the terminal's system directory (error handling code has been removed for clarity):

```
long GetProtectedCounter()
{
DWORD protectedCounter = 0;
CString filename = ::GetSysDir();
filename += _T("system.bin");
CFile file;
file.Open(filename, CFile::modeRead | CFile::modeCreate | CFile::modeNoTruncate);
file.Read(&protectedCounter, sizeof(protectedCounter));
file.Close();
return protectedCounter;
}
```

We believe that the Diebold system violates the FEC requirements by storing the protected counter in a simple, mutable file. By modifying this counter, an adversary could cast doubt on an election by creating a discrepancy between the number of votes cast on a given terminal and the number of votes that are tallied in the election. While the current method of implementing the counter is totally insecure, even a cryptographic checksum would not be enough to protect the counter; an adversary with the ability to modify and view the counter would still be able to roll it back to a previous state. In fact, the only solution that would work would be to implement the protective counter in a tamper-resistant hardware token, but doing so would require physical modifications to existing hardware.

4.2 Tampering with ballot definitions

The "ballot definition" for each election (election.edb) contains everything from the background color of the screen and information about the candidates and issues on the ballot to the PPP username and password to use when reporting the results, if reporting the results over a dial-up connection. This data is neither encrypted nor checksummed (cryptographically or otherwise).

If uninterrupted physical access is *ever* available to the voting terminal after the ballot definition has been loaded, perhaps the night before an election, using a janitor's master keys to the building, then it would be possible for an adversary to tamper with the voting terminals' ballot definition file or to even tamper with the voting software itself. Protections such as physical locks or tamper-evident seals may somewhat allay these concerns, but we would prefer designs that can be robust even against physical tampering.

On a potentially much larger scale, if the voting terminals download the ballot definition over a network connection, then an adversary could tamper with the ballot definition file en-route from the back-end server to the voting terminal; of course, additional poll-worker procedures could be put in place to check the contents of the file after downloading, but we prefer a technological solution. With respect to modifying the file as it is sent over a network, we point out that the adversary need not be an election insider; the adversary could, for example, be someone working at the local ISP. If the adversary knows the structure of the ballot definition, then the adversary can intercept and modify the ballot definition while it is being transmitted. Even if the adversary does not know the precise structure of the ballot definition, many of the fields inside are easy to identify and change, including the candidates' names, which appear as plain ASCII text. Because no cryptographic techniques are in place to guard the integrity of the ballot definition file, an attacker could add, remove, or change issues on the ballot, and thereby confuse the result of the election. In the system, different voters can be presented with different ballots depending on their party affiliations

2This situation can be easily corrected by checking for the FILE ATTRIBUTE TEMPORARY attribute on the directory as described in http://msdn.microsoft.com/library/en-us/wcefiles/htm/_wcesdk_Accessing_Files_on_Other_Storage_Media.asp.

(see CBallotRelSet::Open(), which adds different issues to the ballot depending on the voter's m_ VGroup1 and m_VGroup2 CVoterInfo fields). If an attacker changes the party affiliations of the candidates, then he may succeed in forcing the voters to view and vote on erroneous ballots.3 More subtle

3As an example of what might happen if the party affiliations were listed incorrectly, we note that, according to a news story at http://www.gcn.com/vol19_no33/news/3307-1.html, in the 2000 New Mexico presidential election, over 65,000 votes were incorrectly counted because a worker accidentally had the party affiliations wrong. (We are not claiming this worker had malicious intent, nor are we implying that this error had an effect on the results of the election.)

attacks are also possible. By simply changing the order of the candidates as they appear in the ballot definition, the results file will change accordingly. However, the candidate information itself is not stored in the results file, which merely tracks that candidate 1 got so many votes and candidate 2 got so many other votes. If an attacker reordered the candidates on the ballot definition, voters would unwittingly cast their ballots for the wrong candidate. Ballot reordering attacks would be particularly effective in polling locations known to have more voters of one party than another. (In Section 4.3 and Section 4.5 we consider other ways of tampering with the election results.)

4.3 Impersonating legitimate voting terminals

Consider voting terminals that are configured to upload voting totals to some back-end tabulating authority after an election. An adversary able to pose as a legitimate voting terminal to the tabulating authority could obviously cause (at least temporary) damage by reporting false vote counts to the tabulating system. If the voting terminals use a normal Internet connection, then an adversary with the ability to sniff the connection of a legitimate terminal could learn enough information (e.g., the IP address of the backend server) to be able to impersonate a legitimate terminal. If the terminals use a dialup connection, then the adversary would either need to be able to sniff a legitimate dialup connection to learn the appropriate information (e.g., the dial-up PPP number, login, and password), or must garner that information in another way. The PPP phone number, username, password, and IP address of the back-end server are stored in the registry HKEY_LOCAL_MACHINE\Software\Global Election Systems\AccuVote-TS4\ TransferParams, thus making it easily accessible to an insider working at the polling station. By studying the configuration of the ballot definition files, we learned that the definition files also store the terminal's voting center ID, PPP dial-in number, username, password and the IP address of the back-end server (these are parsed into a CElectionHeaderItem in TSElection\TSElectionObj.cpp). The ballot definition files thus provide another vector for an adversary to learn almost all of the information necessary to impersonate a real voting terminal over a dialup connection (the adversary would also have to create a voting terminal ID, although the ID may or may not be checked for legitimacy by the back-end server).

4.4 Key management and other cryptographic issues with the vote and audit records

Unlike the other data stored on the voting terminal, both the vote records and the audit logs are encrypted and checksummed before being written to the storage device. Unfortunately, neither the encrypting nor the checksumming is done with established, secure techniques. This section summarizes the issues with Diebold's use of cryptography in protecting the vote records and audit logs, and then return to consequences of Diebold's poor choices in subsequent subsections. (Recall that we have already discussed the lack of cryptography in other potions of the system.)

KEY MANAGEMENT. All of the data on a storage device is encrypted using a single, hardcoded DES [22] key:

```
#define DESKEY ((des_key*)"F2654hD4")
```

Note that this value is not a hex representation of a key, nor does it appear to be randomly generated. Instead,

the bytes in the string “F2654hD4” are fed directly into the DES key scheduler. It is well-known that hardcoding
keys into a program’s source code is a bad idea: if the same compiled program image is used on every voting terminal, an attacker with access to the source code, or even to a single program image, could learn the key and thus read and modify voting and auditing records. The case with the Diebold system is even worse: from the CVS logs, we see this particular key has been used without change since December 1998, when the CVS tree for AccuVote-TS version 3 began, and we assume that the key was in use much before
14
that. Although Jones reports that the vendor may have been aware of the key management problems in their code since at least 1997 [16, 17], our findings show that the design flaw was never addressed. The SAIC analysis of Diebold’s system [27] agrees that Diebold needs to redesign their cryptography architecture. The most appropriate solution will likely involve the use of hardware cryptographic coprocessors.
(In a similar fashion, Diebold’s voter, administrator, and ender cards use a hardcoded 8-byte password ED 0A ED 0A ED 0A ED 0A (hexadecimal) to authenticate the voting terminals to the smartcards, transmitted
in cleartext. The smartcards are discussed in Section 3.)
“ENCRYPTION.” Even if proper key management were to be implemented, however, many problems would still remain. First, DES keys can be recovered by brute force in a very short time period [12]. DES should be replaced with either triple-DES [26] or, preferably, AES [8]. Second, DES is being used in CBC mode which requires a random initialization vector to ensure its security. The implementation here always uses zero for its IV. This is illustrated by the call to DesCBCEncrypt in TSElection/RecordFile.cpp;
since the second to last argument is NULL, DesCBCEncrypt will use the all-zero IV.

```
DesCBCEncrypt((des_c_block*)tmp, (des_c_block*)record.m_Data, totalSize,
DESKEY, NULL, DES_ENCRYPT);
```

To correctly implement CBC mode, a source of “strong” random numbers must be used to generate a fresh IV for each encryption [2]. Suitably strong random numbers can be derived from many different sources, ranging from custom hardware to accumulated observations of user behavior.
“MESSAGE AUTHENTICATION.” Before being encrypted, a 16-bit cyclic redundancy check (CRC) of the plaintext data is computed. This CRC is then stored along with the ciphertext in the file and verified whenever the data is decrypted and read. This process in handled by the ReadRecord and WriteRecord
functions in TSElection/RecordFile.cpp. Since the CRC is an unkeyed, public function, it does
not provide any meaningful integrity protection for the data. In fact, by storing it in an unencrypted form, the purpose of encrypting the data in the first place (leaking no information about the contents of the plaintext) is undermined. Standard industry practice would be to first encrypt the data to be stored and then to compute a keyed cryptographic checksum (such as HMAC-SHA1 [1]) of the ciphertext [3, 19]. This cryptographic checksum could then be used to detect any tampering with the plaintext. Note also that each entry has a timestamp, which can be used to detect reordering, although sequence numbers should also be added to detect record deletion.

4.5 Tampering with election results and linking voters with their votes

A likely attack target are the voting records themselves. When stored on the device, the voting records are “encrypted” as described in Section 4.4. If the votes are transmitted to a back-end authority over a network connection, as appears to be the case in at least some areas, no cryptography is used: the votes are sent in cleartext. In particular, CTransferResultsDlg::OnTransfer() writes ballot results to an instance
of CDL2Archive, which then writes the votes in cleartext to a socket without any cryptographic checksum. If the network connection is via a cable modem or a dedicated connection, then the adversary could be an employee at the local ISP. If the voting terminals use a dialup connection directly to the tabulating authority’s network, then the risk of such an attack is less, although still not inconsequential. A sophisticated adversary, e.g., an employee of the local phone company, could tap the phone line and intercept the communication.

TAMPERING WITH ELECTION RESULTS. In Section 4.2 we showed that an adversary could alter election results by modifying ballot definition files, and in Section 4.3 we showed that an adversary could inject fake votes to a back-end tabulating authority by impersonating a legitimate voting terminal. Here we suggest another way to modify the election result: modify the voting records file stored on the device. Because of the poor cryptography described in Section 4.4, an attacker with access to this file would be able to

generate or change as many votes as he or she pleased. Furthermore, the adversary's modified votes would be indistinguishable from the true votes cast on the terminal. The attack described here is more advantageous to an adversary than the attacks in Section 4.2 and Section 4.3 because it leaves no evidence that an attack was ever mounted (whereas the attacks in Section 4.2 and Section 4.3 could be discovered but not necessarily corrected as part of a post-election auditing phase).

If the votes are sent to the back-end authority over a network, then there is another vector for an adversary to modify the election results. Specifically, an adversary with the ability to tamper with the channel could introduce new votes or modify existing votes. Such an attacker could, for example, decrease one candidate's vote count by some number while increasing another's candidate's count by the same number. Of course, to introduce controlled changes such as these to the votes, the attacker would benefit from some knowledge of the structure of the protocol used between the terminals and the back-end server. This form of tampering might later be detected by comparing the memory storage cards to data transmitted across the networks, although the memory storage cards themselves might also be subject to tampering. (We briefly comment that these network attacks could be largely circumvented with the use of standard cryptographic tools, such as SSL/TLS.)

LINKING VOTERS WITH THEIR VOTES. From analyzing the code, we learned that each vote is written *sequentially* to the file recording the votes. This fact provides an easy mechanism for an attacker, such as a poll worker with access to the voting records, to link voters with their votes. A poll worker could surreptitiously track the order in which voters use the voting terminals. Later, in collaboration with other attackers who might intercept the "encrypted" voting records, the exact voting record of each voter could be reconstructed.

If the results are transmitted over a network, as is the case in at least some jurisdictions, then physical access to the voting results is not even necessary. Recall that, when transmitted over the network, the votes are sent in unencrypted, cleartext form.

"RANDOMIZED" SERIAL NUMBERS. While the voter's identity is not stored with the votes, each vote is given a serial number in order to "randomize" the votes after they are uploaded to the back-end tabulating authority. As we noted above, randomizing the order of votes *after* they are uploaded to the the tabulating authority does not prevent the possibility of linking voters to their votes. Nevertheless, it appears that the designers wanted to use a cryptographically secure pseudorandom number generator to generate serial numbers for some post-processing purposes. Unfortunately, the pseudorandom number generator they chose to use (a linear congruential generator) is not cryptographically secure. Moreover, the generator is seeded with static information about the voting terminal and the election.

```
// LCG - Linear Conguential Generator - used to generate ballot serial numbers
// A psuedo-random-sequence generator
// (per Applied Cryptography, by Bruce Schneier, Wiley, 1996)
#define LCG_MULTIPLIER 1366
#define LCG_INCREMENTOR 150889
#define LCG_PERIOD 714025
static inline int lcgGenerator(int lastSN)
{
return ::mod(((lastSN * LCG_MULTIPLIER) + LCG_INCREMENTOR), LCG_PERIOD);
}
```

It is interesting to note that the code's authors apparently decided to use an linear congruential generator because it appeared in *Applied Cryptography* [26] even though in the same work it is advised that such generators should not be used for cryptographic purposes.

4.6 Audit logs

Each entry in a plaintext audit log is simply a timestamped, informational text string. There appears to be no clear pattern for what is logged and what is not. The whole audit log is encrypted using the insecure method described in Section 4.4. An adversary with access to the audit log file could easily change its contents.

At the time that the logging occurs, the log can also be printed to an attached printer. If the printer is unplugged, off, or malfunctioning, no record will be stored elsewhere to indicate that the failure occurred. The following code from TSElection/Audit.cpp demonstrates that the designers failed to consider these issues:

```
if (m_Print && print) {
CPrinter printer;
// If failed to open printer then just return.
CString name = ::GetPrinterPort();
if (name.Find(_T("\\")) != -1)
name = GetParentDir(name) + _T("audit.log");
if (!printer.Open(name, ::GetPrintReverse(), FALSE))
::TSMessageBox(_T("Failed to open printer for logging"));
else {
[ do the printing ]
}
}
```

If the cable attaching the printer to the terminal is exposed, an attacker could create discrepancies between the printed log and the log stored on the terminal by unplugging the printer (or, by simply cutting the cable).

4.7 Attacking the start of an election

Although good election processes would dictate installing the ballot definition files well before the start of the election, we can imagine scenarios in which the election officials must reinstall ballot files shortly before the start of an election, and do not have time to distribute the definition files manually.4

One option for the election officials would be to download the files over the Internet. In addition to the problems we have outlined, we caution against relying on such an approach, as an adversary could mount a traditional Internet denial-of-service attack against the election management's server and thereby prevent the voting terminals from acquiring their ballot definition files in time for the election. Even a general idea of the range of Internet addresses used by the election administration would be sufficient for an attacker to target a large-scale distributed denial of service (DDoS) attack.

Of course, we acknowledge that there are other ways to postpone the start of an election at a voting location that do not depend on Internet DDoS attacks (e.g., flat tires for all poll workers for a given precinct, or other acts of real-world vandalism). Unlike such traditional attacks, however, (1) the network-based attack is relatively easy for anyone with knowledge of the election system's network topology to accomplish; (2) this attack can be performed on a very large scale, as the central distribution point(s) for ballot definitions becomes an effective single point of failure; and (3) the attacker can be physically located anywhere in the Internet-connected world, complicating efforts to apprehend the attacker. Such attacks could prevent or delay the start of an election at all voting locations in a state. We note that this attack is not restricted to the system we analyzed; it is applicable to any system that downloads its ballot definition files using the Internet or otherwise relies upon the Internet.

4In recent elections, we have seen cases where politicians passed away or withdrew from the race very close to the election day.

5 Software engineering

When creating a secure system, getting the design right is only part of the battle. The design must then be securely implemented. We now examine the coding practices and implementation style used to create the voting system. This type of analysis can offer insights into future versions of the code. For example, if a current implementation has followed good implementation practices but is simply incomplete, one would be more inclined to believe that future, more complete versions of the code would be of a similar high quality. Of course, the opposite is also true, perhaps even more so: it is very difficult to produce a secure system by building on an insecure foundation.

Of course, reading the source code to a product gives only an incomplete view into the actions and intentions of the developers who created that code. Regardless, we can see the overall software design, we can read the comments in the code, and, thanks to the CVS repository, we can even look at earlier versions of the code and read the developers' commentary as they committed their changes to the archive.

5.1 Code legacy

Inside cvs.tar we found multiple CVS archives. Two of the archives, AccuTouch and AVTSCE, implement full voting terminals. The AccuTouch code, corresponding to AccuVote-TS version 3, dates from December 1998 to August 2001 and is copyrighted by "Global Election Systems, Inc.," while the AVTSCE code, corresponding to the AccuVote-TS version 4 system, dates from October 2000 to April 2002 and is copyrighted by "Diebold Election Systems, Inc." (Diebold acquired Global Election Systems in September 2001.5) Although the AccuTouch tree is not an immediate ancestor of the AVTSCE tree (from the CVS logs, the AVTSCE tree is actually an import of another AccuTouch-CE tree that we do not have), the AccuTouch and AVTSCE trees are related, sharing a similar overall design and a few identical files. From the comments, some of the code, such as the functions to compute CRCs and DES, date back to 1996 and a company later acquired by Global Election Systems called "I-Mark Systems." We have already remarked (Section 4.4) that the same DES key has been hardcoded into the system since at least the beginning of the AccuTouch tree.

5.2 Coding style

While the system is implemented in an unsafe language6 (C++), the code reflects an awareness of avoiding such common hazards as buffer overflows. Most string operations already use their safe equivalents, and there are comments, e.g., should really use snprintf, reminding the developers to change others.

While we are not prepared to claim that there are no exploitable buffer overflows in the current code, there are at the very least no glaringly obvious ones. Of course, a better solution would have been to write the entire system in a safe language, such as Java or Cyclone [15]. In such a language we would be able to prove that large classes of attacks, including buffer overflows and type-confusion attacks, are impossible assuming a correct implementation of the compiler and runtime system.

Overall, the code is rather unevenly commented. While most files have a description of their overall function, the meanings of individual functions, their arguments, and the algorithms within are more often than not undocumented. An example of a complex and completely undocumented function is the CBallotRelSet::Open function from TSElection/TSElectionSet.cpp as shown in Figure 2.

This block of code contains two nested loops, four complex conditionals, and five debugging assertions, but no comments that explain its purpose. Ascertaining the meaning of even a small part of this code is a huge undertaking. For example, what does it mean for vgroup->KeyId() == -1? That the ID is simply

5http://dallas.bizjournals.com/dallas/stories/2001/09/10/daily2.html

6Here we mean language safety in the technical sense: no primitive operation in any program ever misinterprets data.


```
void CBallotRelSet::Open(const CDistrict* district, const CBaseunit* baseunit,
const CVGroup* vgroup1, const CVGroup* vgroup2)
```

```
{
ASSERT(m_pDB != NULL);
ASSERT(m_pDB->IsOpen());
ASSERT(GetSize() == 0);
ASSERT(district != NULL);
ASSERT(baseunit != NULL);
if (district->KeyId() == -1) {
Open(baseunit, vgroup1);
} else {
const CDistrictItem* pDistrictItem = m_pDB->Find(*district);
if (pDistrictItem != NULL) {
const CBaseunitKeyTable& baseunitTable = pDistrictItem->m_BaseunitKeyTable;
int count = baseunitTable.GetSize();
for (int i = 0; i < count; i++) {
const CBaseunit& curBaseunit = baseunitTable.GetAt(i);
if (baseunit->KeyId() == -1 || *baseunit == curBaseunit) {
const CBallotRelationshipItem* pBalRelItem = NULL;
while ((pBalRelItem = m_pDB->FindNextBalRel(curBaseunit, pBalRelItem))){
if (!vgroup1 || vgroup1->KeyId() == -1 ||
(*vgroup1 == pBalRelItem->m_VGroup1 && !vgroup2) ||
(vgroup2 && *vgroup2 == pBalRelItem->m_VGroup2 &&
*vgroup1 == pBalRelItem->m_VGroup1))
Add(pBalRelItem);
}
}
}
m_CurIndex = 0;
m_Open = TRUE;
}
}
}
```

Figure 2: The function CBallotRelSet::Open function from TSElection/TSElectionSet.cpp. This complex function is completely undocumented.

undefined? Or perhaps that the group should be ignored? Such poorly documented code impairs the ability of both internal developers and external security evaluator to assess whether the code is functioning properly or might lead to a security issue.

5.3 Coding process

An important point to consider is how code is added to the system. From the project's CVS logs, we can see that most recent code updates are in response to specific bugs that needed to be fixed. There are, however, no references to tracking numbers from a bug database or any other indication that such fixes have been vetted through any change-control process. Indeed, each of the programmers7 seem to have completely autonomous authority to commit to any module in the project. The only evidence that we have found that the code undergoes any sort of review comes from a single log comment: "Modify code to avoid multiple exit points to meet Wyle requirements." This refers to Wyle Labs, one of the independent testing authorities charged with certifying that voting machines have met FEC guidelines.

Virtually any serious software engineering endeavor will have extensive design documents that specify how the system functions, with detailed descriptions of all aspects of the system, ranging from the user interfaces through the algorithms and software architecture used at a low level. We found no such documents

in the CVS archive, and we also found no references to any such documents in the source code, despite references to algorithms textbooks and other external sources.

There are also pieces of the voting system that come from third parties. Most obviously, a flaw in the operating system, Windows CE, could expose the system to attack since the OS controls memory manage-

7Through web searches, we have matched each programmer's CVS user names with their likely identities and so can conclude
that they are not group accounts.

ment and all of the device's I/O needs. In addition, an audio library called *fmod* is used.8 While the source to *fmod* is available with commercial licenses, unless this code is fully audited it might contain a backdoor or an exploitable buffer overflow. Since both the operating system and *fmod* can access the memory of the voting program, both must be considered part of the trusted computing base (TCB) as a security vulnerability in either could compromise the security of the voting program itself. The voting terminal's hardware boot instructions should likewise be considered part of the TCB.

Due to the lack of comments, the legacy nature of the code, and the use of third-party code and operating systems, we believe that any sort of comprehensive, top-to-bottom code review would be nearly impossible. Not only does this increase the chances that bugs exist in the code, but it also implies that any of the coders could insert a malicious backdoor into the system without necessarily being caught. The current design deficiencies provide enough other attack vectors, however, that such an explicit backdoor is not required to successfully attack the system. Regardless, even if the design problems are eventually rectified, the problems with the coding process may well remain intact.

Since the initial version of this paper was made available on the Internet, Diebold has apparently "developed, documented, and implemented a change control process" [27]. The details of this revised process have not been made available to the public, so we are unable to comment on their effectiveness.

5.4 Code completeness and correctness

While the code we studied implements a full system, the implementors have included extensive comments on the changes that would be necessary before the system should be considered complete. It is unclear whether the programmers actually intended to go back and remedy all of these issues as many of the comments existed, unchanged, for months, while other modifications took place around them. Of course, while the AVTSCE code we examined appears to have been the current codebase in April 2002, we know nothing about subsequent changes to the code. (Modification dates and locations are easily visible from the CVS logs.) These comments come in a number of varieties. For illustrative purposes, we have chosen to show a few such comments from the subsystem that plays audio prompts to visually-impaired voters.

• Notes on code reorganization:

```
/* Okay, I don't like this one bit. Its really tough to tell where m AudioPlayer
should live. [...] A reorganization might be in order here. */
```

• Notes on parts of code that need cleaning up:

```
/* This is a bit of a hack for now. [...] Calling from the timer message
appears to work. Solution is to always do a 1ms wait between audio clips. */
```

• Notes on bugs that need fixing:

```
/* need to work on exception *caused by audio*. I think they will currently
result in double-fault. */
```

There are, however, no comments that would suggest that the design will radically change from a security perspective. None of the security issues that have been discussed in this paper are pointed out or marked for correction. In fact, the only evidence at all that a redesign might at one point have been considered comes from outside the code: the Crypto++ library9 is included in another CVS archive in cvs.tar. However, the library was added in September 2000, before the start of the AVTSCE AccuVote-TS version 4 tree, and appears to have never been used. (The subsequent SAIC [27] and RABA [24] analyses report that many

of the problems we identify are still applicable to recent versions of the AccuVote-TS system, implying
8http://www.fmod.org/
9http://www.eskimo.com/~weidai/cryptlib.html

that, at least up to the version that SAIC and RABA analyzed, there has not been any radical change to the AccuVote-TS system.)

6 Conclusions

Using publicly available source code, we performed an analysis of the April 2002 snapshot of Diebold's AccuVote-TS 4.3.1 electronic voting system. We found significant security flaws: voters can trivially cast multiple ballots with no built-in traceability, administrative functions can be performed by regular voters, and the threats posed by insiders such as poll workers, software developers, and janitors is even greater. Based on our analysis of the development environment, including change logs and comments, we believe that an appropriate level of programming discipline for a project such as this was not maintained. In fact, there appears to have been little quality control in the process.

For quite some time, voting equipment vendors have maintained that their systems are secure, and that the closed-source nature makes them even more secure. Our glimpse into the code of such a system reveals that there is little difference in the way code is developed for voting machines relative to other commercial endeavors. In fact, we believe that an open process would result in more careful development, as more scientists, software engineers, political activists, and others who value their democracy would be paying attention to the quality of the software that is used for their elections. (Of course, open source would not solve all of the problems with electronic elections. It is still important to verify somehow that the binary program images running in the machine correspond to the source code and that the compilers used on the source code are non-malicious. However, open source is a good start.) Such open design processes have proven successful in projects ranging from very focused efforts, such as specifying the Advanced Encryption Standard (AES) [23], through very large and complex systems such as maintaining the Linux operating system. Australia is currently using an open source voting system10.

Alternatively, security models such as the voter-verified audit trail allow for electronic voting systems that produce a paper trail that can be seen and verified by a voter. In such a system, the correctness burden on the voting terminal's code is significantly less as voters can see and verify a physical object that describes their vote. Even if, for whatever reason, the machines cannot name the winner of an election, then the paper ballots can be recounted, either mechanically or manually, to gain progressively more accurate election results. Voter-verifiable audit trails are required in some U.S. states, and major DRE vendors have made public statements that they would support such features if their customers required it. The EVM project11 is an ambitious attempt to create an open-source voting system with a voter-verifiable audit trail—a laudable goal.

The model where individual vendors write proprietary code to run our elections appears to be unreliable, and if we do not change the process of designing our voting systems, we will have no confidence that our election results will reflect the will of the electorate. We owe it to ourselves and to our future to have robust, well-designed election systems to preserve the bedrock of our democracy.

Acknowledgments

We thank Cindy Cohn, David Dill, Badri Natarajan, Jason Schultz, Tracy Volz, David Wagner, and Richard Wiebe for their suggestions and advice. We also thank the state of Maryland for hiring SAIC and RABA and the state of Ohio for hiring Compuware to independently validate our findings.

10http://www.elections.act.gov.au/EVACS.html
11http://evm2003.sourceforge.net

References

[1] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In

N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, Berlin Germany, Aug. 1996.
[2] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 394–403. IEEE Computer Society Press, 1997.
[3] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer-Verlag, Berlin Germany, Dec. 2000.
[4] California Internet Voting Task Force. *A Report on the Feasibility of Internet Voting*, Jan. 2000. http://www.ss.ca.gov/executive/ivote/.
[5] *Voting: What Is; What Could Be*, July 2001. http://www.vote.caltech.edu/Reports/.
[6] D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy*, 2(1):38–47, 2004.
[7] Compuware Corporation. *Direct Recording Electronic (DRE) Technical Security Assessment Report*, Nov. 2003. http://www.sos.state.oh.us/sos/hava/files/compuware.pdf.
[8] J. Daemen and V. Rijmen. *The Design of Rijndael: AES–The Advanced Encryption Standard*. Springer-Verlag, Berlin Germany, 2002.
[9] Diebold Election Systems. AVTSCE source tree, 2003. http://users.actrix.co.nz/dolly/Vol2/cvs.tar.12
[10] D. L. Dill, R. Mercuri, P. G. Neumann, and D. S. Wallach. *Frequently Asked Questions about DRE Voting Systems*, Feb. 2003. http://www.verifiedvoting.org/drefaq.asp.
[11] Federal Election Commission. *Voting System Standards*, 2001. http://fecweb1.fec.gov/pages/vss/vss.html.
[12] J. Gilmore, editor. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. O'Reilly, July 1998.
[13] D. Gritzalis, editor. *Secure Electronic Voting*. Springer-Verlag, Berlin Germany, 2003.
[14] B. Harris. *Black Box Voting: Vote Tampering in the 21st Century*. Elon House/Plan Nine, July 2003.
[15] T. Jim, G. Morrisett, D. Grossman, M. Hicks, J. Cheney, and Y. Wang. Cyclone: A safe dialect of C. In *USENIX Annual Technical Conference*, June 2002.
[16] D. W. Jones. *Problems with Voting Systems and the Applicable Standards*, May 2001. Testimony before the U.S. House of Representatives' Committee on Science, http://www.cs.uiowa.edu/˜jones/voting/congress.html.
12The cvs.tar file has been removed from this website.
22
[17] D. W. Jones. *The Case of the Diebold FTP Site*, July 2003. http://www.cs.uiowa.edu/˜jones/voting/dieboldftp.html.
[18] A. Kerckhoffs. *La Cryptographie Militaire*. Libraire Militaire de L. Baudoin & Cie, Paris, 1883.
[19] H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 310–331. Springer-Verlag, Berlin Germany, 2001.
[20] R. Mercuri. *Electronic Vote Tabulation Checks and Balances*. PhD thesis, University of Pennsylvania, Philadelphia, PA, Oct. 2000.
[21] National Science Foundation. *Report on the National Workshop on Internet Voting: Issues and Research Agenda*, Mar. 2001. http://news.findlaw.com/cnn/docs/voting/nsfe-voterprt.pdf.
[22] NBS. Data encryption standard, January 1977. Federal Information Processing Standards Publication 46.

[23] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, and E. Roback. *Report on the Development of the Advanced Encryption Standard (AES)*, Oct. 2000.
[24] RABA Innovative Solution Cell. *Trusted Agent Report: Diebold AccuVote-TS Voting System*, Jan. 2004. http://www.raba.com/press/TA_Report_AccuVote.pdf.
[25] A. D. Rubin. Security considerations for remote electronic voting. *Communications of the ACM*, 45(12):39–44, Dec. 2002. http://avirubin.com/e-voting.security.html.
[26] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, New York, second edition, 1996.
[27] Science Applications International Corporation. *Risk Assessment Report: Diebold AccuVote-TS Voting System and Processes*, Sept. 2003. http://www.dbm.maryland.gov/SBE.

Priedas Nr. 2

WINvote™ Wireless Functionality Improves Efficiency, Reduces Costs
Touch-screen voting has become the technology of choice by election administrators. However, not all systems are alike; in fact they vary greatly.

The WINvote™ system possesses features and functionality that will potentially revolutionize the Election Equipment and Solutions Industry.

The functionality linchpin of the WINvote™ system is its wireless LAN (IEEE 802.11b) system - called the Wireless Information Network (WIN) -- that enables the user to communicate remotely with the major components of the voting system.

Two examples are: (a.) **WINstage**, a exclusive warehouse management system that is being hailed as the most beneficial cost-reduction tool available in the industry, and (b.) **WINmanager**, an equally exclusive precinct management system designed to make polling places more efficient and reduce procedural anomalies.

Winning with WINvote
The WINvote™ system is expected to become the next industry standard by which all voting systems are measured for the following reasons: (a.) it offers productivity tools to aid in substantial cost reductions never before available to election administrators, (b.) it simplifies procedures throughout the entire process, and (c.) it embodies technology that election processes and procedures that are sure to evolve over time.

Priedas Nr. 3

Name/Model: Unilect Patriot
Vendor: Unilect Corporation

Brief Description:

The Unilect Patriot is a multilingual electronic voting system where the voter presses onscreen to indicate his/her preference. Election officials program ballot information at a central location, load election data into an “InfoPack” which is then inserted into a Precinct Control Unit (see photo). Individual terminals are connected together into the PCU to receive ballot data.

Voters make selections by pressing the box surrounding a candidate’s name, navigating through ballot pages by means of navigation buttons, review their ballots by means of a summary screen, and can go back and change selections before casting their final votes. Vote data is stored in redundant memory inside each terminal. After polls close, vote data is loaded back into the PCU and can then be transmitted via modem to a tabulation center. Alternatively, the InfoPack” (which store the vote data from all of the terminals at the polling place) can be removed from the PCU and taken to a tabulation center.

Detailed Machine and Voting Process Information:

Pre-election procedures

Ballot information is generally programmed by city or county election officials and later delivered to polling places. In order to create the ballots that will be used on the Patriot terminals, a menu-driven program is used to prompt the entry of all offices, candidates and propositions in order to "code" the election. Election officials have the option to print out the ballot on paper for proofreading purposes.

Ballot data is transferred to polling places in the form of an administrator interface loaded with precinct-specific data. By placing an "InfoPack" (a little larger than a pack of cigarettes) into the "InfoPacket" attached to the PC, the necessary ballot instructions are electronically transferred from the election supervisor PC to the InfoPack (about 5 seconds). It is then inserted into the Precinct Control Unit (“PCU” – see above photo) for the appropriate precinct. It is tested, sealed and sent to the precinct along with the prescribed number of Patriot Voting Devices.

In order to load the proper ballots into the Patriot terminals, the precinct workers place the PCU on the table and set up each booth. These are then connected from one to another by a cord similar in size to a lamp cord. The PCU is then turned on.

With the PCU in front of them, one of the precinct workers breaks the seal on the "Open Polls" latch, slides it open, and touches the red button underneath. This immediately causes the printer to print a report showing all the candidates with zero totals.

Voting on election day

The polls are now open and await the first voter. As each voter is checked in, the precinct worker determines which ballot style the voter is entitled to use (only if a split precinct, or a certain party in a primary), and assigns them to a particular open Voting Device booth. Only the offices on which the voter is entitled to vote should be displayed on the Voting Device. If a voter is judged to be a "Provisional" or "Challenged" or "Conditional" or "Affidavit" voter, a special button allows him/her to vote on the Patriot. Depending on local procedures, this ballot is not normally counted on election evening, but is added later if found to be valid.

Each voter should see the ballot electronically displayed on the screen. In localities where more than one language is required, the voter has the option of choosing which language he wishes to view. Usually the entire ballot will take more than one screen, so the voter

may move forward or back by touching the appropriate box on the screen. The voter makes each candidate selection by touching anywhere in the box containing that name. As each is selected, that candidate's box becomes highly illuminated and a red “x” is placed next to the name. If a mistake is made, the selected candidate's box may be touched again (de-selecting him), and the new candidate selected. The entire ballot may be reviewed at any time to check who was previously selected.

Write- ins may be electronically entered by touching the "Write-In" box for a particular office. Upon selection of the write-in option, the screen changes to display an alphabet, and the write-in name may be spelled by touching the proper letters.

When completely finished voting, the "Review Choices" area is touched. This will automatically display on a single screen all choices made by that voter. It will also highlight those offices which were not completed by the voter. At that point, the voter may either "Make Ballot Changes" or "Record Ballot Now". At this point, the voter has completed his/her task and leaves.

Accessibility features

Unilect offers several accessibility features associated with the Patriot, although it is unclear if these features are readily available and installed on each terminal. These include the ability to disconnect and move terminals to enable “curb-side” voting, headphones and different shaped response buttons to facilitate voting by the sightimpaired, etc.

Preferential/Proportional voting

The Patriot Voting System also allows the use of "Preferential" or "Proportional" voting where permitted. This allows each voter to rank their candidates in order of desirability. Candidates are then elected by quota. (Such voting methods or used by some jurisdictions to eliminates the need for additional runoff elections.

Post-election procedures

At the end of the election day, a seal is broken on the "Close Polls" latch, opened and an exposed red button is touched. Several copies of the final precinct report are automatically printed, showing the candidates and their vote totals. Presumably, vote data is at this stage transferred back from the terminals to the PCU. Where a standard telephone line is available, the line is inserted into the phone jack of the Precinct Control Unit. All precinct totals are then transmitted directly to the Central Office PC. It is unclear whether this process is automated or whether poll workers must take affirmative action to initiate this upload. Another seal is then broken and the InfoPack is removed in order to be taken to a tabulation center.

At the tabulation center, each InfoPack (for precinct totals not sent by telephone) is inserted into a central PC equipped with the Patriot InfoPacket for a five (5) seconds to load the totals into the PC. Throughout election evening, summary reports can be printed showing all of the up-to-the- minute totals as they are received (including all write-ins).

Equipment – at the polling place

Patriot voting devices (5 to 8 lbs.), including a standard punch card type voting booth (13 lbs.) that folds up into an attache case. The Voter Unit may contain either a 10.4" (diagonal measurement) black and white screen or a 15" color screen with 256 available colors.

One precinct control unit (PCU) per precinct (30 lbs.), has an election worker control panel covering all aspects of running the activities in the precinct, a printer which allows the printing of precinct results as soon as the polls close, a battery to assure proper operation when "wall" electricity becomes unavailable, an InfoPack which contains the

brains of the ballot as well as final vote totals, and an internal modem for direct transfer of totals from a standard telephone in the precinct to the Patriot Central Station, in the election office.

Equipment – at the election office

One Patriot Central Station (per jurisdiction), includes a PC or PC network powerful enough to code all ballots in the jurisdiction, accept all totals directly from the precincts via modem, and/or accept totals directly from InfoPacks and absentee ballots, instantly adds and tallies together including in-person or early voting ballots, and disseminates that information via summary reports throughout election evening (as well as individual precinct reports, canvass, logs and other miscellaneous reports). Other equipment included are InfoPackets, a printer (to match the jurisdiction's needs), modems, and an absentee ballot reader.

Priedas Nr. 4

Name/Model: AVC Edge
Vendor: Sequoia Voting Systems, Inc.

Brief Description:

The Sequoia AVC Edge is a voter-activated multilingual touchscreen system that records votes on internal flash memory. Voters insert a "smart-card" into the machine and then make their choices by touching an area on a computer screen, much in the same way that modern ATMs work. The votes are then recorded to internal electronic flash memory. When polls close, the votes for a particular machine are written to a “PCMCIA card” which are removed from the system and either physically transported to election headquarters or their contents transmitted via computer network.

How To Vote On This Machine:

When the voter enters the precinct, he or she is given a "smart-card" by a poll worker after confirming the voter is registered. A "smart-card" – a card the size and shape of a credit-card – contains a computer chip, some memory and possibly basic data such as the voter's political party. The voter then takes the smart-card to a voting machine and inserts the smart-card into the yellow slot visible in the middle picture above. The first screen presented to the voter is one that allows him or her to choose the ballot language. After using the touchscreen to vote, 1) the record of the vote is directly recorded electronically to two flash memory cards and 2) the voter's smart card is reset to ensure that the voter can only vote once. The AVC Edge may also be equipped in some precincts to print a voter-verified paper audit trail using the VeriVote printer. In this case, the voter will inspect the printout which is displayed underneath glass. If the paper accurately reflects the vote, the voter indicates so using the touchscreen and casts the vote; the printed paper is withdrawn into the machine to protect privacy. If the paper is incorrect, the voter may mark it as spoiled and change his or her vote using the touchscreen interface. After the vote is cast, the smart-card pops out of the machine and the voter returns it to a poll worker.

When the polls close, a poll worker or election official inserts a different-type of smart card, an administrator card, into each voting machine and puts the machine into a postelection mode where it will no longer record votes. At this point, the machine writes the votes from its internal memory to flash memory on a "PCMCIA card.” The PCMCIA card is merely a removable form of flash memory. A printed tape of all votes cast or vote totals for the voting machine can also be printed out at this time depending on local procedure and regulations.

The PCMCIA cards are removed from each machine and either taken to a central tabulation facility or to remote tabulation facilities. At the tabulation facility the votes are copied from the PCMCIA cards and into a central computer database where precincts are combined to result in an aggregate vote. The votes may also be transmitted to the central tabulation facility via a closed "Intranet", the Internet or modem. The PCMCIA cards and possible any printouts from the voting machines can then become part of the official record of the election.

Past Problems

- June 2004: New Jersey. In Morris County, the central tabulation system could not read the data from the PCMCIA cards. The system showed zeros.1
- November 2003: California. After a battery problem occurred during the election in Santa Clara County, Sequoia technicians worked on the machines without oversight from county officials. Following November's election in Santa Clara County, Sequoia sent over a group of technicians to make adjustments to voting

machines that experienced battery problems.2

- November 2002: New Mexico. In Bernalillo County, 48,000 people voted early but no race showed more than 36,000 votes. The cause was a software bug.3
- April 2002: Florida. In Hillsborough County, one precinct could not transfer data on 24 out of 26 PCMCIA cards. Results summaries were faxed in and entered by hand.4 In March 2003, a similar problem plagued 2 out of 678 PCMCIA cards.5
- March 2002: Florida. In Palm Beach County much went wrong. When voters selected their language, the Edge froze up. Other reports indicate votes registering for wrong candidate.6 15 PCMCIA cards were temporarily lost and central system would not report result. In a race won by 4 votes, 78 were blank; voters reported erratic machine behavior.7
- November 2000: California. During the 2000 presidential election in Riverside County, a computer from Sequoia began dropping touch-screen ballots from the vote tally. A Sequoia salesman who was on hand intervened and fixed the problem.8

1 "Montville and Chatham mayors ousted." NEW JERSEY STAR-LEDGER, June 9, 2004.
2 “Electronic voting's hidden perils.” SAN JOSE MERCURY NEWS. February 1, 2004.
3 "Election results certified after software blamed." ALBUQUERQUE TRIBUNE, November 19, 2002.
4 "Officials still searching for election glitch: The new system could not send the tabulations to the elections office." ST. PETERSBURG TIMES, April 6, 2002.
5 "Elections Chief Sees Nearly Flawless Vote." ST. PETERSBURG TIMES, March 5, 2003.
6 "Human goofs, not machines, drag vote tally into next day." PALM BEACH POST, 14 March 2002.
7 "Out of Touch: You press the screen. The machine tells you that your vote has been counted. But how can you be sure?" NEW TIMES, April 24, 2003.
8 *Id*., note 2.

Priedas Nr. 5

AVC ADVANTAGE

The AVC Advantage represents the culmination of over a century of experience and technological innovation in the voting system industry. A 100% direct recording, electronic voting machine, the AVC Advantage is easy to use and offers the highest level of accuracy and security.

AVC ADVANTAGE IS EASY TO OPERATE

• Easily maneuvered by just one person, the AVC Advantage is quick and easy to set up.
• Messages and prompts are displayed throughout all phases of operation.

AVC ADVANTAGE IS EASY TO VOTE

• The tactile voting switch with visual indicator provides positive feed-back to the voter.
• A large candidate area accommodates large type for easy to read ballots.
• The AVC Advantage is completely wheelchair accessible with no election officer intervention.

AVC ADVANTAGE IS EASY TO MAINTAIN

• No electronics knowledge is required to operate or maintain the AVC Advantage.
• Error messages are easy to understand for quick and simple troubleshooting.
• The AVC Advantage is designed with modular components for easy part replacement or system upgrade.

AVC Advantage

AVC ADVANTAGE IS FAST AND RELIABLE

• Immediate copies may be printed of Results and Audit-Trail Reports.
• A visual display of vote totals serves as a backup.
• Transportable memory cartridges are compatible with the Sequoia Pacific central system.

AVC ADVANTAGE EARNS YOUR CONFIDENCE

• Diagnostics are performed automatically at every power up.
• Logic and Accuracy tests are required before, and can be optionally performed after, the election.
• Continual background testing verifies correct system operation and data integrity.
• The AVC Advantage stores an electronic randomized record of all votes cast. This Audit Trail can be printed on demand.
• Operator logs document all system activity during the pre-election, election, and post-election cycle.

Immediate copies may be printed of Results and Audit-Trail Reports.

Operator panels can be conveniently mounted on either side of the AVC Advantage for ease of use by one election worker.

What's the advantage of AVC Advantage?

Vote processing via multiple independent data paths, randomized and stored three different ways to assure absolute secrecy of the vote, absolute accuracy in vote

counting, and absolute verifiability of results. Nothing less than the complete elimination of human error. Nothing less than the best. Nothing less than AVC Advantage.

Features & Benefits

CONFIRMATION OF VOTER SELECTIONS

Tactile switches, visual indicators and the LCD message display provide confirmation to the voter that the AVC Advantage has correctly recorded the voter's selections.

HANDICAP ACCESSIBILITY

Easy adjustment required for wheel-chair voters.

ELECTRONIC WRITE-INS

Eliminates interpreting voter intent. Write-in votes are recorded electronically and stored redundantly in the AVC Advantage in the Cartridge. The write-in votes are transferred automatically to the central system, the Election Database System. The write-in votes for the jurisdiction may then be printed by contest for recounts.

PRIVACY CURTAIN

Completely encloses the voting area, ensuring total voting secrecy.

FULL FACE BALLOT DISPLAY

Voting is quicker and simpler than paper ballot or paginating systems. Does not allow over voting. Eliminates spoiled or rejected ballots.

MESSAGE DISPLAYS

Prompts the Pollworker and Voter throughout the operation of the AVC Advantage.

CONSOLIDATION

Allows machines within a precinct to be automatically accumulated and totals printed at the polling site.

BATTERY BACK-UP

Built in 16+ hour battery provides uninterrupted use of the AVC Advantage. Power switches from AC to DC.

FOUR LARGE RUBBER CASTERS WITH FIVE INCH CLEARANCE. Easy to deliver the AVC Advantage machines to the polling site. Easy for one Pollworker to move within the polling site.

PRE-ELECTION LOGIC AND ACCURACY TESTS

A mandatory function during election preparation for ballot verification and public oversight of ballot integrity.

POST-ELECTION LOGIC AND ACCURACY TESTS

An optional function after the election for ballot verification and public oversight of ballot integrity.

VOTE SIMULATION

Allows the AVC Advantage to automatically conduct high

volume vote tests.

100% ACCURACY

Redundant storage of ballots and totals which are updated and verified between each voter.

FEC CERTIFIED

You can have confidence in the quality of design and components that go into each AVC Advantage. No system has been more publicly and thoroughly examined and tested than the AVC Advantage.

AUDIT*TRAIL

Provides an unalterable, randomized electronic record (ballot image) of all votes cast during an election. The ballot image is redundantly stored in the AVC Advantage and in the Results Cartridge. A chronological Operators Log records the time, date and nature of all significant system events. The Audit Trail and Operator Log may be printed after the election on demand.

BACKGROUND AND POWER UP DIAGNOSTICS.

Provide continual verification of system integrity.

EARLY VOTING

Each AVC Advantage supports over 2,000 precincts to accommodate jurisdiction-wide early voting on a single machine.

EXPANDABLE BALLOT SIZE

Allows jurisdiction to purchase only the number of voting positions they will need (252, 336, 420, or 504 positions). Expansion to 504 positions can be easily accommodated.

FULL SERVICE AND SUPPORT

Highly trained technical support staff and rapid shipment of component modules.

POOL PA RTS PROGRAM FOR ELECTRONIC BOARD REPA I R

Assures the jurisdiction of long-term supply of available electronic parts at a reduced price over purchasing new parts.

LOW ON-GOING OPERATING COSTS

The AVC Advantage is manufactured with quality components. It has a history of long, useful life with minimal service and maintenance required. Set-up is accomplished in-house, and requires only one ballot per machine, not a ballot for each voter. Ballot can be generated by Sequoia’s central system and printed on a plotter in-house.

SPECIFICATIONS:

STORAGE POSITION SIZE

LENGTH: 46.5 IN

DEPTH: 24.5 IN

HEIGHT: 39.3 IN

VOTING POSITION SIZE

LENGTH: 46.5 IN

DEPTH: 55.0 IN

HEIGHT: 75.3 IN
WEIGHT: 225-265 lbs.
depending upon system configuration.
POWER
Operates on a 110 VAC with system backup battery capacity of over 16 hours, depending upon system configuration.

NEWADDRESS
www.sequoiavote.com

Priedas Nr. 6

eSlate™ Electronic Voting System

Hart InterCivic's eSlate electronic voting solution is the most fully-featured, affordable and accurate Direct Record Electronic (DRE) system available today. Successfully used in the November presidential election, eSlate has won acclaim for its ease of use, accessibility to disabled voters, and fast, efficient ballot tabulation.

No other system can match eSlates performance, accountability and flexibility in ballot types. eSlate's unique Precision Ballot Navigation System™ avoids problems typically associated with touch screen systems and ensures voters can confidently and accurately register their votes. All eSlate components are designed and manufactured under strict ISO 9001 certified quality management standards.

eSlate 3000
The eSlate 3000 has a flexible ballot presentation, durable polycarbonate screen, integrated selector and is secure and affordable.

Judge's Booth Controller™ (JBC 1000)
eSlate's JBC 1000s manage the election process in the precinct. The JBC 1000 controls up to 12 eSlate 3000s and enables the election judge to know which booths are in use at any given time.

Disabled Access Unit™ (DAU 5000)
The eSlate system is ADA accessible by design. The eSlate 3000 can be upgraded to a DAU 5000 to accommodate various devices that support voting by the disabled. Key features incorporated into the eSlate's ADA compliant electronic voting system through the DAU 5000 include:

- Special interfaces for the physically challenged, including head movement switches and "sip and puff" switches (that allow severely physically impaired voters to cast their ballot using only their breath).
- An audio ballot reader to support visually impaired voters, including audible signals that provide confirmation with each selection.
- A simple navigation method that is modeled after systems commonly used by the disabled.

All disability features can be used interchangeably, in whole or part, with the eSlate's standard interfaces, allowing the voter to overcome whatever challenges he or she might face in casting his or her vote.

Press Release

Mobile Ballot Box™ (MBB)
eSlate's PCMCIA flash memory card is the storage medium for all voting information to operate the eSlate system. No batteries are required to securely store election data.

Ballot Origination Software System™ (BOSS)
BOSS enables users to define and create ballot styles for all precincts. Election data is written to MBBs and

will configure every product of the eSlate system in any location.

Tally™

Tally accepts results from and tabulates all Early Voting tabulation, Absentee tabulation, Election Day tabulation and Election Canvass information. Tally has reporting flexibility as it contains standard reports as well as a custom report writer to produce customized reports for specific jurisdictions.

Ballot Now™

Ballot Now Digital Ballot Imaging services allow ballots to be produced as needed by the customer. Returned mailed ballots are processed using commercially available scanners providing a fully scalable solution.

Voter Registration

Hart's eSlate system has an optional voter registration component. The eSlate system is integrated and fully compatible with the VEMACS™ voter registration system from VOTEC and other voter registration applications based on relational database architecture.

Priedas Nr. 7

Name/Model: iVotronic
Vendor: Election Systems & Software (ES&S)

Brief Description:

ES&S' iVotronic Touch Screen Voting System is a poll worker-activated, portable, multilingual touchscreen system that records votes on internal flash memory. A poll worker uses a device called a Personal Electronic Ballot (PEB; pictured above at left) to turn the machine on and enable voting. Voters choose their ballot language and then make their selections using a touchscreen, much in the same way that modern ATMs work. When the polls close, poll workers move summary data from each machine onto the PEB. The PEBs are then transported to election headquarters or their contents transmitted via a computer network.

Detailed Voting Process:

When the voter enters the polling place, a poll worker first confirms the voter is registered. Then the poll worker walks with the voter to an iVotronic and inserts the PEB in the PEB slot (visible as the rectangular slot in the upper left corner of the middle image above). The PEB communicates with the iVotronic using infrared signals, much like a TV remote control works, except that the PEB and iVotronic will not communicate unless the PEB is completely inserted. If the election requires a a specific ballot style, the poll worker chooses this for the voter. Activation by the PEB enables the iVotronic to vote once.

The voter then selects a ballot language and makes decisions using the touchscreen. When the voter is done, he or she presses a small “vote” button at the very top of the iVotronic to cast the vote. The vote is then recorded to three internal flash memories that reside inside the machine. A fourth memory is a removable card, called a “compact flash” (CF) card; note that CF is the same technology used in many digital cameras to store photos. During the election, the CF card holds audio files (for those with visual disabilities) and ballot definitions; vote data is written to the CF card when the machine is closed.

A poll worker closes the polls by using the PEB with a password to enter a supervisor menu on each iVotronic. After closing the election for a given machine, summary vote data are transmitted to the PEB via infrared signals.1 After the PEB is used to close all the iVotronic machines, it contains all the summary data for the precinct. Depending on local regulations and procedures, poll workers can use a “printer pack” at this point to print the result summary from the PEB on to paper. The PEB for that precinct, any printouts and the CF cards are then either phys ically transported to a central tabulation facility, and in addition, the data may be transmitted by telephone using the modem included in the printer pack.

All of the electronic ballot images and event log data remain in the iVotronic until it is cleared for the next election. Many jurisdictions use the serial port on the back of the iVotronic to extract this data for archival storage during normal post-election procedures. This data duplicates what is stored on the compact flash card, and some jurisdictions save only this data or only the data from the compact flash card.

Past Problems:

- January 2004: Florida. In a special election for the State House District 91 seat, with only one

1 Note that the vote data transmitted to the PEB at the closing of a machine is summary vote data instead of raw vote data;
that is, it is a summary of the votes recorded rather than each individual electronic ballot as stored inside the iVotronic's
internal memory. In order to do a proper recount or error analysis, one would need to remove the CF cards from the
iVotronics and seal the CF cards for a precinct with the PEB and any printouts. This information is courtesy of Doug
Jones of the University of Iowa.

item on the ballot, ES&S electronic voting machines showed a total of 134 undervotes – that is, 134 ballots in which voters did not select a candidate even though it was a single-race election. The winner received 12 more votes than the runner- up. Florida law requires a manual recount of invalid votes when the winning margin is less than one-quarter of one percent. However, election officials determined that no recount was required because the 134 invalid votes were cast on electronic voting machines, and there is no record of the original votes.2

- May 2003: Florida. An internal review of election results by a Miami- Dade county election official found that a DRE system sold by ES&S and used in the May 20, 2003 North Miami Beach runoff election (as well as in earlier elections) was "unusable" for auditing, recounting or certifying an election due to a "serious bug" in the software.3 As of August 2004, the newest software releases from ES&S fix this bug, which turns out to have been triggered by a low battery condition.4
- November 2002: North Carolina. At two early- voting locations in Wake County, North Carolina (Raleigh), iVotronics failed to record 436 ballots. This was due to a problem in the firmware of the machines.5 Firmware is a kind of software loaded on read-only memory so that it cannot be easily changed.
- October 2002: Texas. Democrats said they received several dozen complaints from people who said that they selected a Democratic candidate but that their vote appeared beside the name of a Republican on the screen. Some votes cast for Republicans were counted for Democrats.6
- September 2002: Florida. A spot check of machines revealed two problems. First, several Miami- Dade precincts, each with hundreds of voters, are listed as showing one or even no votes cast on election day. Second, differences arose within the same precincts between vote totals produced by the main tabulation system and a backup system.7

2 "Electronic Vote Recount Stumps Broward Officials." SUN-SENTINEL, January 10, 2004.
3 "Count Crisis? Election Officials Warn of Glitches that May Scramble Vote Auditing." MIAMI DAILY BUSINESS
REVIEW, May 16, 2004. "Glitch Forces Change in Vote Audits." THE MIAMI HERALD, May 15, 2004.
4 Doug Jones, personal communication. Note that the software that fixes this bug has made it through ITA testing and state
testing in at least Florida.
5 "Electronic Ballots Fail To Win Over Wake Voters, Election Officials; Machines Provide Improper Vote Count At Two
Locations," WRAL-TV RALEIGH-DURHAM, Nov. 2, 2002.
6 "Area Democrats say early votes miscounted," THE DALLAS MORNING NEWS, Oct. 22, 2002.
7 "Leahy: Unskilled workers to blame," MIAMI HERALD, Sept. 12, 2002.

Priedas Nr. 8

PostgreSQL

PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES, Version 4.2, developed at the University of California at Berkeley Computer Science Department. POSTGRES pioneered many concepts that only became available in some commercial database systems much later.

PostgreSQL is an open-source descendant of this original Berkeley code. It supports SQL92 and SQL99 and offers many modern features:

- complex queries
- foreign keys
- triggers
- views
- transactional integrity
- multiversion concurrency control

Additionally, PostgreSQL can be extended by the user in many ways, for example by adding new

- data types
- functions
- operators
- aggregate functions
- index methods
- procedural languages

And because of the liberal license, PostgreSQL can be used, modified, and distributed by everyone free of charge for any purpose, be it private, commercial, or academic.

Advantages

PostgreSQL offers many advantages for your company or business over other database systems.

Immunity to over-deployment

Over-deployment is what some proprietary database vendors regard as their #1 licence compliance problem. With PostgreSQL, no-one can sue you for breaking licensing agreements, as there is **no associated licensing cost for the software.**

This has several additional advantages:

- More profitable business models with wide-scale deployment.
- No possibility of being audited for license compliance at any stage.
- Flexibility to do concept research and trial deployments without needing to include additional licensing costs.

Better support than the proprietary vendors

In addition to our strong support offerings, we have a vibrant community of PostgreSQL professionals and enthusiasts that your staff can draw upon and contribute to.

Significant saving on staffing costs

Our software has been designed and created to have much lower maintenance and tuning requirements than the leading proprietary databases, yet still retain all of the features, stability, and performance.

In addition to this our training programs are generally regarded as being far more cost effective, manageable, and practical in the real world than that of the leading proprietary database vendors.

Legendary reliability and stability

Unlike many proprietary databases, it is extremely common for companies to report that PostgreSQL has never, ever crashed for them in several years of high activity operation. Not even once. It just works.

Extensible

The source code is available to all at no charge. If your staff have a need to customise or extend PostgreSQL in any way then they are able to do so with a minimum of effort, and with no attached costs. This is complemented by the community of PostgreSQL professionals and enthusiasts around the globe that also actively extend PostgreSQL on a daily basis.

Cross platform

PostgreSQL is available for almost every brand of Unix (34 platforms with the latest stable release), and Windows compatibility is available via the Cygwin framework. Native Windows compatibility is also available with version 8.0 and above.

Designed for high volume environments

We use a multiple row data storage strategy called MVCC to make PostgreSQL extremely responsive in high volume environments. The leading proprietary database vendor uses this technology as well, for the same reasons.

GUI database design and administration tools

Several high quality GUI tools exist to both administer the database (pgAdmin, pgAccess) and do database design (Tora, Data Architect).

Technical Features

- Fully ACID compliant.
- ANSI SQL compliant.
- Referential Integrity.

- Replication (non-commercial and commercial solutions) allowing the duplication of the master database to multiple slave machines.
- Native interfaces for ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python, and Ruby.
- Rules.
- Views.
- Triggers.
- Unicode.
- Sequences.
- Inheritance.
- Outer Joins.
- Sub-selects.
- An open API.
- Stored Procedures.
- Native SSL support.
- Procedural languages.
- Hot stand-by (commercial solutions).
- Better than row-level locking.
- Functional and Partial indexes.
- Native Kerberos authentication.
- Support for UNION, UNION ALL and EXCEPT queries.
- Loadable extensions offering SHA1, MD5, XML, and other functionality.
- Tools for generating portable SQL to share with other SQL-compliant systems.
- Extensible data type system providing for custom, user-defined datatypes and rapid development of new datatypes.
- Cross-database compatibility functions for easing the transition from other, less SQL-compliant RDBMS.

A Brief History of PostgreSQL

The object-relational database management system now known as PostgreSQL is derived from the POSTGRES package written at the University of California at Berkeley. With over a decade of development behind it, PostgreSQL is now the most advanced open-source database available anywhere.

The Berkeley POSTGRES Project

The POSTGRES project, led by Professor Michael Stonebraker, was sponsored by the Defense Advanced Research Projects Agency (DARPA), the Army Research Office (ARO), the National Science Foundation (NSF), and ESL, Inc.

The implementation of POSTGRES began in 1986. The initial concepts for the system were presented in The design of POSTGRES and the definition of the initial data model appeared in The POSTGRES data model. The design of the rule system at that time was described in *The design of the POSTGRES rules system.* The rationale and architecture of the storage manager were detailed in The design of the POSTGRES storage system.

POSTGRES has undergone several major releases since then. The first "demoware" system became operational in 1987 and was shown at the 1988 ACM-SIGMOD Conference. Version 1, described in The implementation of POSTGRES, was released to a few external users in June 1989. In response to a critique of

the first rule system (A commentary on the POSTGRES rules system), the rule system was redesigned (On Rules, Procedures, Caching and Views in Database Systems) and Version 2 was released in June 1990 with the new rule system. Version 3 appeared in 1991 and added support for multiple storage managers, an improved query executor, and a rewritten rule system. For the most part, subsequent releases until Postgres95 focused on portability and reliability.

POSTGRES has been used to implement many different research and production applications. These include: a financial data analysis system, a jet engine performance monitoring package, an asteroid tracking database, a medical information database, and several geographic information systems. POSTGRES has also been used as an educational tool at several universities. Finally, Illustra Information Technologies (later merged into Informix, which is now owned by IBM.) picked up the code and commercialized it. In late 1992, POSTGRES became the primary data manager for the Sequoia 2000 scientific computing project.

The size of the external user community nearly doubled during 1993. It became increasingly obvious that maintenance of the prototype code and support was taking up large amounts of time that should have been devoted to database research. In an effort to reduce this support burden, the Berkeley POSTGRES project officially ended with Version 4.2.

Postgres95

In 1994, Andrew Yu and Jolly Chen added a SQL language interpreter to POSTGRES. Under a new name, Postgres95 was subsequently released to the web to find its own way in the world as an open-source descendant of the original POSTGRES Berkeley code.

Postgres95 code was completely ANSI C and trimmed in size by 25%. Many internal changes improved performance and maintainability. Postgres95 release 1.0.x ran about 30-50% faster on the Wisconsin Benchmark compared to POSTGRES, Version 4.2. Apart from bug fixes, the following were the major enhancements:

- The query language PostQUEL was replaced with SQL (implemented in the server). Subqueries were not supported until PostgreSQL (see below), but they could be imitated in Postgres95 with user-defined SQL functions. Aggregate functions were re-implemented. Support for the GROUP BY query clause was also added.
- In addition to the monitor program, a new program (psql) was provided for interactive SQL queries, which used GNU Readline.
- A new front-end library, libpgtcl, supported Tcl-based clients. A sample shell, pgtclsh, provided new Tcl commands to interface Tcl programs with the Postgres95 server.
- The large-object interface was overhauled. The inversion large objects were the only mechanism for storing large objects. (The inversion file system was removed.)
- The instance-level rule system was removed. Rules were still available as rewrite rules.
- A short tutorial introducing regular SQL features as well as those of Postgres95 was distributed with the source code
- GNU make (instead of BSD make) was used for the build. Also, Postgres95 could be compiled with an unpatched GCC (data alignment of doubles was fixed).

PostgreSQL

By 1996, it became clear that the name "Postgres95" would not stand the test of time. We chose a new name, PostgreSQL, to reflect the relationship between the original POSTGRES and the more recent versions with SQL capability. At the same time, we set the version numbering to start at 6.0, putting the numbers back into the sequence originally begun by the Berkeley POSTGRES project.

The emphasis during development of Postgres95 was on identifying and understanding existing problems in the server code. With PostgreSQL, the emphasis has shifted to augmenting features and capabilities, although work continues in all areas.

Priedas Nr. 9

Affordable uniprocessor server for small businesses and multi-location enterprises

IBM eserver® xSeries 206

njoy features that
mplify deployment
nd improve usability
and maintenance

- *New IBM ServeRAID™-7e enables you to install and configure RAID-0 or -1 without additional adapters*
- *New systems management functions include ASF 2.0, providing secure, remote power—on and off control*

Highlights

Take advantage of innovative server technology such as new 64-bit processing performance

Leverage flexible configurations and availability on demand for distributed environments and growing business needs

The IBM _____ ® xSeries® 206 offers value, innovation and usability at affordable prices for small businesses and workgroups. The x206 features Intel® Extended Memory 64 Technology supporting an easy migra- tion path to 64-bit computing, delivering investment protection and enhanced performance. New features help control IT costs and accommodate growth.

Affordable innovation

Help lower total computing costs:

- *New simple-swap Serial ATA drives can be connected and disconnected easily*

Flexibility for growth

Configure an x206 with features that enable future growth. New Intel® Pentium® 4 processors double the speed of previous server generations. Hot-swap SCSI models increase availability as your needs grow. And support for the Remote Supervisor Adapter II helps increase availability with virtual network control as much as 24x7.

Easy to deploy, easy to run

Standard on the x206 is integrated ServeRAID-7e enabling RAID-0 and -1. And the improved server design provides ease of access without compromising security.

Priedas Nr. 10

Building Networks for People

D-Link ANT24-1801

High Gain Directional Yagi Antenna

- **18 dBi Signal Gain**
- **2.4GHz Frequency Range**
- **Directional Orientation for Precise Wireless Signals**
- **Includes Cable that Converts N-female to Reverse SMA**
- **Made with Weatherproof and Corrosion Resistant Material**

Outdoor 18 dBi High Gain Directional Yagi Antenna

The D-Link ANT24-1801 connects to the DWL-900AP+, DI-614+, DI-714P+, DWL-900AP, DI-714 and DI-713P to extend the range of coverage for the wireless network.

The D-Link ANT24-1801 antenna provides extended coverage for an existing 802.11b wireless local area network (WLAN). The D-Link ANT24-1801 comes with a conversion cable that allows connection directly to the D-Link DWL-900AP+ wireless access point, DI-614+ wireless router, DI-714P+ wireless router, DWL-900AP wireless access point, DI-714 wireless router and the DI-713P wireless router Rev C1 or later wireless broadband router.

This product includes the following items: Mounting Kit, Lightning Surge Protector, Conversion Cable.

The D-Link ANT24-1801 requires an access point or wireless broadband gate- ways with a reverse SMA connector. The D-Link ANT24-1801 comes with a con- version cable that allows connection directly to the DWL-900AP+ wireless access point, DI-614+ wireless router, DI-714P+ wireless router, DWL-900AP wire- less access point, DI-714 wireless router and the DI-713P wireless router Rev C1
or later wireless broadband router.

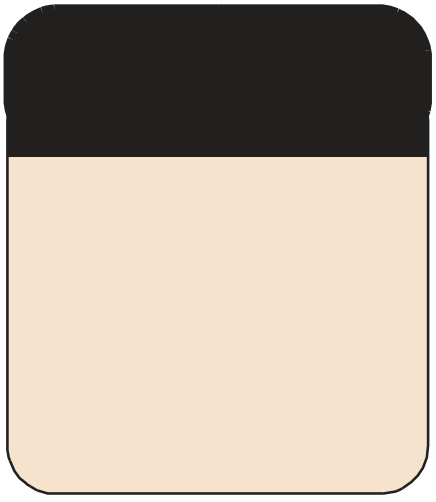
Note: To obtain optimal results in extend- ing wireless range with outdoor antenna installations, it is recommended that pro- fessional installer service is consulted for site survey and and proper installation.

Antenna Manual

Mounting Kit

Conversion Cable

Lighting Surge Protector



Piedas Nr. 11

Business and Enterprise Solutions

Layer 2 Switch

Web-Smart 24-Port 10/100/1000BASE-T With 2 Combo SFP Expansion Slots

DGS-1224T

FEATURES

High Performance Wire-Speed Architecture:

-48Gbps Switching Capacity
-Non-Blocking Architecture
-8,000 MAC Address Table
-2 Hardware Priority Queue
-Port-based QoS
-Port-based VLAN
-Link Aggregation
-Port Mirroring
-Port Setting (Speed, Availability, Flow Control)
-Web-based Management
-Auto Discovery Utility

Operational Simplicity:

-True Plug & Play
-Auto-Negotiation
-Clear, At A Glance Per Port LED Indicators
-Auto MDI/MDI-X Detection

Investment Protection:

-Support for Industry Standards
-10/100/1000Mbps Migration
-Fiber Media Support Via SFPs

The DGS-1224T blends plug-&-play simplicity with exceptional performance and reliability, to create a cost-effective solution for bandwidth-starved workgroups and departments. With (24) 10/100/1000BASE-T ports & (2) combo SFP expansion slots, the DGS-1224T is designed to help provide a simple and economical migration path from 10BASE-T or 100BASE-TX to 1000BASE-T Gigabit. Key features of the DGS-1224TG include:

- High Performance Wire-Speed Architecture
- Operational Simplicity
- L2 features including QoS for bandwidth sensitive applications and link aggregation
- Remote Management via Web Browser and Auto Discovery Utility
- Investment Protection

High Performance Wire-Speed Architecture-

The DGS-1224T delivers superior performance with exceptional value. With a switching capacity of up to 48Gbps and full duplex wire-speed forwarding, the DGS-1224T is an ideal solution for the most demanding bandwidth intensive applications. With support for 8,000 MAC addresses, the DGS-1224T can be used as a cost-effective Gigabit wiring closet solution or a high performance backbone aggregation device.

Operational Simplicity-

The DGS-1224T supports a wide array of plug-in-play features:

- 10/100/1000Mbps auto-negotiation
- Universal UTP cable recognition for auto straight-through or crossover cable detection
- Full/Half duplex support
- IEEE standards support

Investment Protection-

The DGS-1224T helps to protect customers' existing and future network infrastructure investments by:

- Work with unmanaged Gigabit switches along with competitive price point and features
- Providing a plug-&-play migration for existing 10/100BASE-TX devices to 1000BASE-T
- Standards-based feature support to help insure multi-vendor compatibility
- Fiber Gigabit Media Support Via 2 Combo SFP Expansion Slots

The D-Link DGS-1224T is a flexible, high performance, and value-oriented solution for a wide range of Gigabit applications.

D-Link®
Building Networks for People

Priedas Nr. 12

D-Link®
Building Networks for People

DI-524

G
802.11g
54Mbps

2.4 GHz

BUILT-IN FIREWALL

FREE 24/7 TECH SUPPORT
(USA ONLY)*

AirPlus™ G 802.11g/2.4GHz Wireless Router

Protect & Share Your Internet Connection

Up to 54Mbps[1] and Compatible with 11g and 11b

Advanced Firewall & Parental Control Increased Security Controls

Built-In 4-Port Switch

Quick and Easy Setup

offers 4 Ethernet Ports to support multiple computers.

The advanced wireless technology built into the DI-524 offers a maximum wireless signal rate of up to 54Mbps[1] through its wireless channels allowing streaming videos and other high bandwidth applications, such as online gaming events, to operate without the hassle of Ethernet cables. The ability to use high bandwidth applications also makes streaming real time programs more enjoyable and more efficient.

Network Security is a precautionary threat and with the DI-524's built-in advanced firewall, these threats

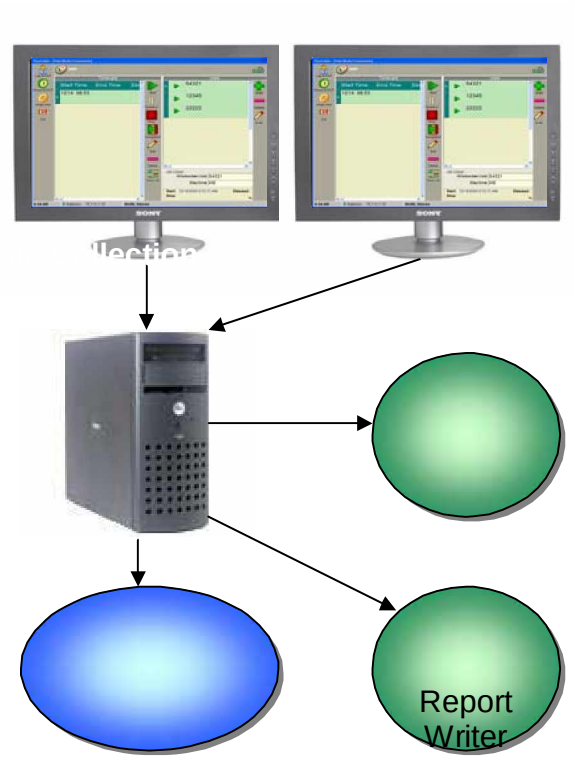
D-Link, the industry leader in wireless networking, introduces another breed of wireless router. The D-Link AirPlus™ G series of high speed devices are capable of transferring maximum wireless signal rate of up to 54Mbps[1] in the 2.4GHz frequency — the same wireless frequency as 802.11b. The D-Link DI-524 also

are minimized, making it more difficult for hackers to penetrate through. Some firewall features includes functions to allow or disallow certain ports to be open for certain applications. Time scheduling can be applied to the firewall rules to have specific ports open at certain times and to be closed at other times. Features like content filtering, MAC filtering, URL blocking, and Domain blocking are useful tools to prevent other unwanted intruders from connecting to your network or browsing restricted sites.

The easy to use configuration wizard takes only minutes to setup and guides users step-by-step through configuring the DI-524. With all these features and an user-friendly utility, the DI-524 provides an enhanced networking experience.

[1] Maximum wireless signal rate based on IEEE Standard 802.11g specifications. Actual data throughput will vary. Network conditions and environmental factors lower actual data throughput rate.

Priedas Nr. 13

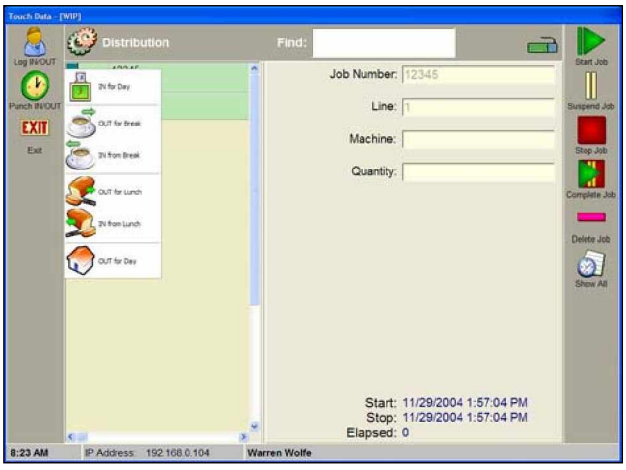


The Touch Data™ system is the answer to your need for easy, accurate touch screen data collection. Designed with the manufacturing process in mind, Touch Data™ utilizes a touch screen for easy, efficient data entry.

Your information is displayed to each employee in real time, allowing them to choose from your open "jobs" or "work orders", filtered by work center, department, employee, etc. The employee can see their work in real time! No need to key in numbers, they can select from a correct list.

Features

- Touch screen data entry
- Optional badge reader, barcode guns and fingerprint biometric authentication.
- Configurable, flexible workflow process
- The ability for one employee to work on multiple jobs concurrently, with selections to divide the labor
- You can define the prompts and data selection options by
  - **Job** – each job can be specifically defined for special prompts, data display, data validation and behavior
  - **Terminal** - each touch screen terminal can be specifically defined for prompts, data display, data validation and behavior
  - **Global** - prompts, data lookups, data validation and behavior can be defined globally for all terminals and jobs
- Open Database – MS SQL Server database
- Incremental searches – as data is entered or scanned, Touch Data™ will automatically jump down to the closest match in the validation list
- Automatic Dropdown Lists – on entry of the data field, Touch Data™ can display valid selections in a dropdown list automatically
- Validation lists – can show valid selections according to
  - Specific field selections
  - Specific field selections dependent on prior fields data
- Full virtual keyboard
- Display "currently active" jobs or "all" jobs
- Color and icon coded job list for easy recognition of job status:
  - **Green** – Active and in-progress
  - **Yellow** – Active but "Suspended"
  - **Silver** – Job "Stopped"
  - **Red** – Job "Completed"
- Automatic Data capture with accurate data including:
  - Employee
  - Terminal that it was captured from
  - Status of the job
  - Detailed data of the job
  - Transaction type
  - Transaction START and END date and time
  - Job completion status



Open architecture and the ability to integrate with Legiant Web Timecard time and attendance system, Touch Data™ is the

answer to your shop floor data collection requirements. You will have the ability to see, in real time, WIP, where an order is in production, work center utilization, which employees have worked on a job, work centers the job has traveled through, and to see if you are over or under production standards in real time. The Touch Data™ system can automatically feed data to your ERP system as well as calculate real time labor cost and feed payroll.

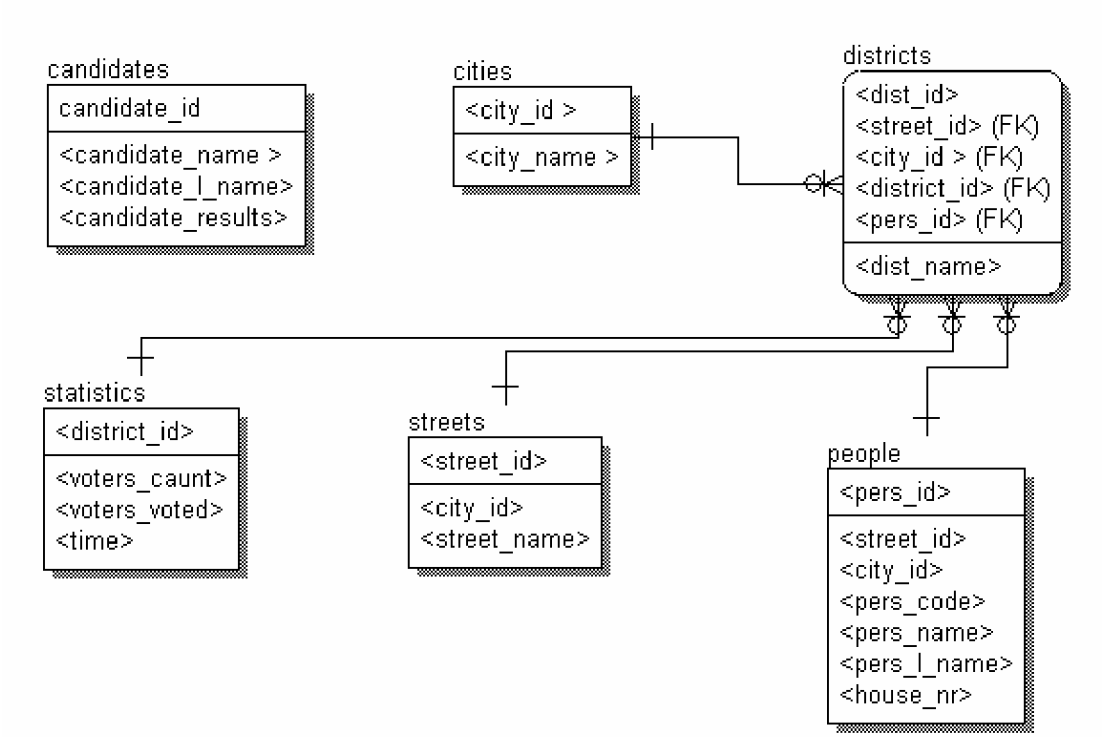
Priedas Nr. 14

Duomenų bazės modelis

Viena svarbiausių E-rinkimų taikomosios sistemos dalis - duomenų bazė. Duomenų bazę galima perskirti į dvi pagrindines dalis: duomenys susiję su rinkėjais ir rinkiminėmis apygardomis, bei informacija apie kandidatų surinktus balsus. Paveikslas Nr.6

Centrinės darbo stoties duomenų bazės modelis:

Loginė schema:



1 pav. Centrinės darbo stoties duomenų bazės loginė schema

Duomenų bazę sudaro duomenys, reikalingi rinkėjų sąrašams bei pažymėjimams generuoti, bei statistikos ir rezultatams saugoti. Duomenų bazę sudaro lentelės :

candidates – lentelė, sauganti kandidatų sarašą; ją sudaro laukai:

- *candidate_id* – unikalus numeris, identifikuojantis kandidatą. Kiekvienas kandidatas atpažįstamas pagal tai, kokį identifikacinį numeri jis turi. Tai padeda išvengti klaidų ir optimizuoti infirmaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *1209877*.
- *candidate_name* – kandidato pilnas vardas. Vardo ilgis neribojamas, norint išvengti klaidų įvedinėjant ilgus kandidatų vardus. Negali būti tuščias. *Text* tipo. Pvz.: *Petras*.
- *candidate_l_name* – kandidato pilna pavardė. Pavardės ilgis neribojamas, norint išvengti klaidų įvedinėjant ilgas kandidatų pavardes. Negali būti tuščias. *Text* tipo. Pvz.: *Jonaitynas*.

- *candidate_results* – kandidato surinktų balsų kiekis rinkimų metu. Pradinė reikšmė – *0. Integer* tipo. Pvz.: *150*.

cities – lentelė, sauganti miestų sąrašą; ją sudaro laukai:

- *city_id* – unikalus numeris, identifikuojantis miestą. Kiekvienas miestas atpažįstamas pagal tai, kokį identifikacinį numerį jis turi. Tai padeda išvengti klaidų ir optimizuoti informaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *25698*.
- *city_name* – miesto pilnas pavadinimas. Pavadinimo ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Kaunas*.

streets – lentelė, kurioje saugomas gatvių sąrašas, suskirstytas pagal miestus; ją sudaro laukai:

- *street_id* - unikalus numeris, identifikuojantis gatvę. Kiekviena gatvė atpažįstama pagal tai, kokį identifikacinį numerį jis turi. Tai padeda išvengti klaidų ir optimizuoti informaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *123597*.
- *city_id* –numeris, rodantis kuriame mieste yra ši gatvė. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *198987*.
- *street_name* – pilnas gatvės pavadinimas. Pavadinimo ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Donelaičio*.

districts – lentelė, kurioje saugoma informacija apie apylinkes; ją sudaro laukai:

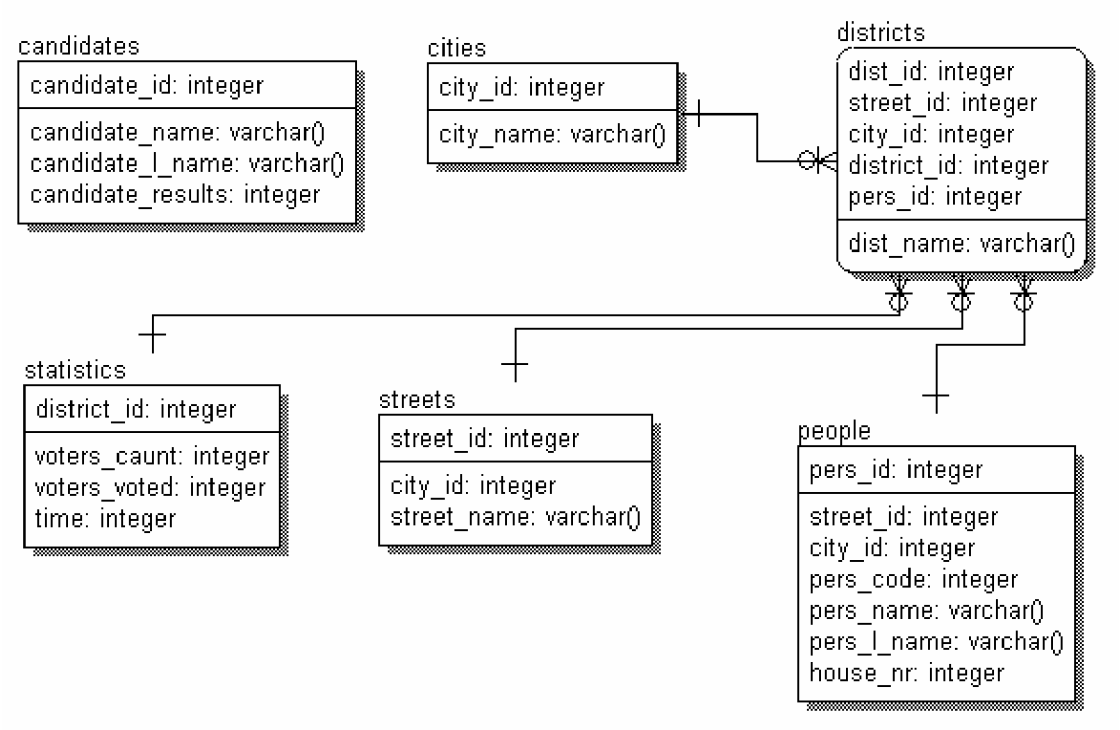
- *dist_id* – unikalus numeris, identifikuojantis apylinkę. Kiekviena apylinkė atpažįstama pagal tai, kokį identifikacinį numerį ji turi. Tai padeda išvengti klaidų ir optimizuoti informaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *123564*.
- *city_id* – numeris, rodantis, kuriame mieste yra ši apylinkė. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *165877*.
- *street_id* – numeris, rodantis, kurioje gatvėje yra ši apylinkė. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *13548*.
- *dist_name* – pilnas apylinkės pavadinimas. Pavadinimo ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Varpo*.

people – lentelė, kurioje saugoma informacija apie rinkėjus; ją sudaro laukai:

- *pers_id* – unikalus numeris, identifikuojantis rinkimuose dalyvausiantį asmenį. Kiekvienas asmuo atpažįstamas pagal tai, kokį identifikacinį numerį jis turi. Tai padeda išvengi klaidų ir optimizuoti informaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *312894*.
- *street_id* – numeris, rodantis kurioje gatvėje gyvena rinkėjas. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *3187*.
- *city_id* – numeris, rodantis kuriame mieste gyvena rinkėjas. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *54987*.
- *pers_code* – lauke talpinamas asmens kodas, kuris reikalingas kaip papildomas identifikatorius. Unikalus. Negali būti tuščias. *Integer* tipo. Pvz.: *38005140615*.

- *pers_name* – pilnas rinkimuose dalyvaujančio asmens vardas. Vardo ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Jonas*.
- *pers_l_name* – pilna rinkimuose dalyvaujančio asmens pavardė. Pavardės ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Petraitis*.
- *house_nr* – pilnas rinkimuose dalyvaujančio asmens adresas, reikalingas kaip papildomas identifikatorius, be to rūšiuojant rinkėjus pagal apylinkes. Negali būti tuščias. *Integer* tipo. Pvz.: *22*.

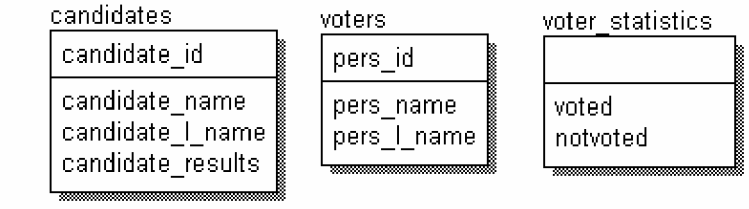
Fizinė schema:



2 pav. Centrinės darbo stoties duomenų bazės fizinė schema

Apygardos duomenų bazės modelis:

Loginė schema:



3 pav. Apygardos duomenų bazės loginė schema

Apygardos duomenų bazę sudaro duomenys apie kandidatus ir rinkėjus, saugomi dvejose lentelėse:

c*andidates* – tai lentelė kurioje saugomi kandidatų duomenys, ja sudaro šie laukai:

- *candidate_id* – unikalus numeris, identifikuojantis kandidatą. Kiekvienas kandidatas atpažįstamas pagal tai, kokį identifikacinį numerį jis turi. Tai padeda išvengti klaidų ir optimizuoti informaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *123456*.

- *candidate_name* – kandidato pilnas vardas. Vardo ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Petras*.
- *candidate_l_name* – kandidato pilna pavardė. Pavardės ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Petraitis*.
- *candidate_results* – kandidato surinktų balsų kiekis rinkimų metu. Pradinė reikšmė – *0. Integer* tipo. Pvz.: 250.

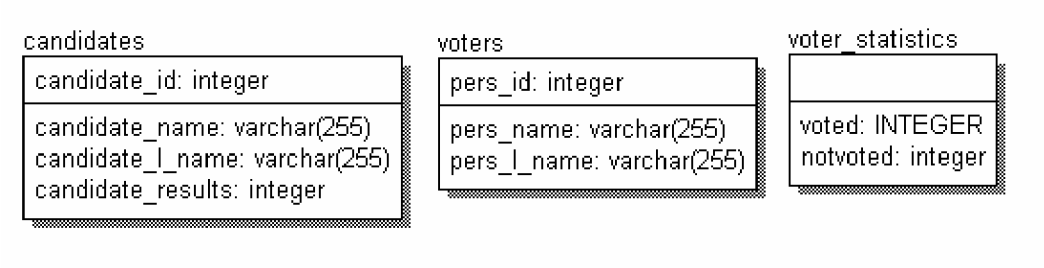
voters - lentelė, kurioje saugoma informacija apie rinkėjus; ją sudaro laukai:

- *pers_id* – unikalus numeris, identifikuojantis rinkimuose dalyvausiantį asmenį. Kiekvienas asmuo atpažįstamas pagal tai, kokį identifikacinį numerį jis turi. Tai padeda išvengi klaidų ir optimizuoti informaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *312894*.
- *pers_name* – pilnas rinkimuose dalyvaujančio asmens vardas. Vardo ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Jonas*.
- *pers_l_name* – pilna rinkimuose dalyvaujančio asmens pavardė. Pavardės ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Petraitis*.

voter_statistics – lentelė, kurioje renkama statistinė informacija apie balsavusių ir nebalsavusių rinkėjų skaičių:

- *voted – integer* tipo laukas. Pradinė reikšmė 0. Kai siunčiama statistika iš apylinkių, lauke sumuojamos balsavusių rinkėjų reikšmės. Gavus informaciją iš visų apylinkių ir nusiuntus ją centrinei darbo stočiai, lauko reikšme keičiama į 0 ir laukiama sekančio statistikos siuntimo ciklo.
- *notvoted – integer* tipo laukas. Pradinė reikšmė 0. Kai siunčiama statistika iš balsavimo terminalų, lauke sumuojamos dar nebalsavusių rinkėjų reikšmės. Gavus informaciją iš visų terminalų ir nusiuntus ją centrinei darbo stočiai, lauko reikšme keičiama į 0 ir laukiama sekančio statistikos siuntimo ciklo.

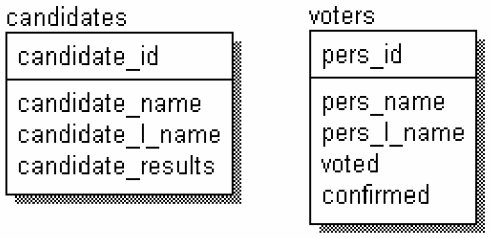
Fizinė schema:



4 pav. Apygardos duomenų bazės fizinė schema

Balsavimo terminalo duomenų bazės modelis:

Loginė schema:



5 pav. Balsavimo terminalo duomenų bazės loginė schema

Balsavimo terminalo duomenų bazę sudaro duomenys apie kandidatus ir rinkėjus, saugomi dvejose lentelėse:

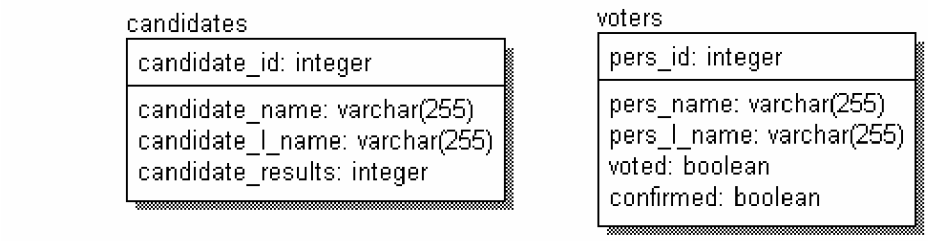
c*andidates* – tai lentelė kurioje saugomi kandidatų duomenys, ja sudaro šie laukai:

- *candidate_id* – unikalus numeris, identifikuojantis kandidatą. Kiekvienas kandidatas atpažįstamas pagal tai, kokį identifikacinį numeri jis turi. Tai padeda išvengti klaidų ir optimizuoti infirmaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *123456*.
- *candidate_name* – kandidato pilnas vardas. Vardo ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Petras*.
- *candidate_l_name* – kandidato pilna pavardė. Pavardės ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Petraitis*.
- *candidate_results* – kandidato surinktų balsų kiekis rinkimų metu. Pradinė reikšmė – *0. Integer* tipo. Pvz.: 250.

voters - lentelė, kurioje saugoma informacija apie rinkėjus; ją sudaro laukai:

- *pers_id* – unikalus numeris, identifikuojantis rinkimuose dalyvausiantį asmenį. Kiekvienas asmuo atpažįstamas pagal tai, kokį identifikacinį numerį jis turi. Tai padeda išvengi klaidų ir optimizuoti informaciją. Negali būti tuščias. *Integer* tipo. Raktinis lentelės laukas. Pvz.: *312894*.
- *pers_name* – pilnas rinkimuose dalyvaujančio asmens vardas. Vardo ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Jonas*.
- *pers_l_name* – pilna rinkimuose dalyvaujančio asmens pavardė. Pavardės ilgis ribojamas 255 simboliais, siekiant sumažinti atminties poreikį. Negali būti tuščias. *Varchar* tipo. Pvz.: *Petraitis*.
- *voted* – *boolean* tipo laukas. Galimos reikšmės Taip arba Ne. Pradinė reikšmė Ne. Jei rinkėjas balsuoja, reikšmė keičiama į Taip. Laukas reikalingas, kad žinotume ar rinkėjas jau balsavo.
- *confirmed* – *boolean* tipo laukas. Galimos reikšmės Taip arba Ne. Pradinė reikšmė Ne. Jei rinkėjas atėjo į apylinkę balsuoti ir užsiregistravo įrašoma reikšmė Taip, tada jis jau gali eiti balsuoti. Laukas reikalingas pradinei rinkėjų registracijai apylinkėje.

Fizinė schema:



6 pav. Balsavimo terminalo duomenų bazės fizinė schema