

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Darius Gudavičius

**Neįgalių vaikų abilitacijos  
informacinė sistema**

Magistro darbas

Darbo vadovas

doc. dr. Vigintas Šakys

Kaunas, 2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Darius Gudavičius

**Neįgalių vaikų abilitacijos  
informacinė sistema**

Magistro darbas

Recenzentas  
doc. dr. Antanas Lenkevičius

2007-01-09

Vadovas  
Šakys

2007-01-09

Atliko  
IFM-1/4 gr. stud.  
Darius Gudavičius

2007-01-09

. doc. dr. Vigintas

Kaunas, 2007

# Turinys

<b>Summary .....</b>	<b>5</b>
<b>Įvadas.....</b>	<b>6</b>
<b>1. Sistemos paskirtis .....</b>	<b>9</b>
1.1. Sistemos kūrimo pagrindas.....	9
1.2. Sistemos tikslai .....	9
1.3. Organizacijos veiklos analizė.....	10
1.4. Tyrimo sritis, objektas ir problema .....	13
1.5. Išvados.....	15
<b>2. Sistemos analizė .....</b>	<b>16</b>
2.1. Analoginės sistemos.....	16
2.2. Analizės metodų, priemonių parinkimas .....	19
2.3. Neįgalųjų poreikiai sistemai.....	20
2.3.1. Vartotojo sąsaja .....	20
2.3.2. Pritaikomos įvedimo sistemos .....	21
2.4. Projektavimo metodų, priemonių parinkimas .....	25
2.5. Išvados.....	25
<b>3. Reikalavimų informacinei sistemai specifikacija.....</b>	<b>27</b>
3.1. Reikalavimų modelis.....	27
3.2. Vartotojai.....	27
3.2.1. Reikalavimai vartotojams .....	27
3.2.2. Vartotojų tipai ir teisės .....	28
3.2.3. Vartotojų autorizavimas .....	33
3.3. Reikalavimai sistemai.....	34
3.3.1. Funkciniai sistemos reikalavimai .....	34
3.3.2. Nefunkciniai sistemos reikalavimai.....	35
3.4. Diegimo aplinka .....	36
3.5. Reikalavimai operacinei sistemai.....	37
3.5.1. Kliento operacinė sistema.....	37
3.5.2. Serverio operacinė sistema .....	39
3.6. Reikalavimai įrangai .....	39
3.7. Kritiniai sistemos faktoriai .....	39

3.8.	<i>Išvados</i> .....	40
<b>4.</b>	<b>Sistemos aprašas</b> .....	<b>41</b>
4.1.	<i>Techninė užduotis</i> .....	41
4.1.1.	Varotojo sąsaja .....	41
4.1.2.	Operacinės sistemos .....	41
4.1.3.	Funkcionalumas.....	42
4.1.4.	Autentifikavimas .....	42
4.1.5.	Realizavimas.....	43
4.2.	<i>Sistemos projektas</i> .....	43
4.2.1.	Architektūra .....	43
4.2.2.	Vizualus sistemos modelis .....	43
4.2.3.	Funkcijos ir funkcionalumas .....	50
4.2.4.	Dalykinės srities klasių diagrama.....	51
4.2.5.	Veikimo principai ir sekų diagramos .....	53
4.2.6.	Programavimo kalbos .....	56
4.2.7.	Duomenų struktūros .....	61
4.3.	<i>Bendradarbiaujančios sistemos</i> .....	66
4.4.	<i>Rizikos įvertinimas</i> .....	66
4.5.	<i>Sistemos komponentai ir įdiegimas</i> .....	68
<b>5.</b>	<b>Testavimas</b> .....	<b>70</b>
5.1.	<i>Testavimo etapai</i> .....	70
5.2.	<i>Testavimo planas ir rezultatai</i> .....	72
5.3.	<i>Išvados</i> .....	74
	<b>Išvados</b> .....	<b>75</b>
	<b>Terminų ir santraukų žodynas</b> .....	<b>77</b>
	<b>Literatūros sąrašas</b> .....	<b>78</b>
	<b>Priedai</b> .....	<b>80</b>

## Summary

### Information system for children with disabilities

The purpose of this project is to help children with disabilities to talk through special cards using available new information technologies. These kids are from ages 8 till 15. I collaborated with KVAC(Kauno Vaikų Abilitacijos Centras) and there met children who cant talk normally because of their physical condition, however their mental condition is quite evolved or even sterling for their age. These special cards are created by Norwegian doctors especially for this kind of disability and for people with such mental and physical condition. In KVAC children used them for very long time. These cards are made from paper and colored in black and white. Each of these cards represents emotion, feelings, things, occupations and many others real word objects. So my solution was to move these cards to computer and adapt them for better usage and also to make kind of chat where more than one person can be involved and naturally would help person with disabilities. Main advantages of this software is that it can be used over LAN or thru the Internet there is no difference, also new technologies like AJAX let the system perform faster(saving bandwidth) then other available technologies. Besides there is no complex installation process, system is ready to work immediatly without requiring detailed customization. Also one of the main thing why I developed this system is because of lack of such type of software over Lithuania and even over the world.

## Įvadas

Komunikacija – vienas iš svarbiausių požymių išskyres žmogų iš gyvūnų pasaulio. Tik bendraujant išlaikomas, perduodamas kitoms kartoms sukauptas patyrimas, išreiškiami jausmai, emocijos, norai bei siekiai.

Vienas iš pačių pagrindinių ir būtinausių žmonių poreikių yra bendravimas. Neįsivaizduojame savęs negalėdami suprasti kito. Aklieji bendrauja Brailio raštu, kurtieji - rankų gestų kalba. Tačiau kai kuriems žmonėms net toks bendravimas yra per daug sudėtingas tai žmonės, kurie nebendrauja dėl įvairių fizinių, protinių ir kitų sutrikimų. Komunikacijos sutrikimai nevienodi jie vyrauja nuo dalinai sunkaus bendravimo iki visiško negalėjimo bendrauti. Vaikai negalėdami kalbėti arba negalėdami tam atlikti tikrų judesių savo norams išreikšti dažniausiai visiškai nebendrauja. Dirbant su tokio pobūdžio vaikais, specialistų taikomas neverbalinis bendravimas (ženklai, gestai, ideogramos, kt.).

Psichiškai ir fiziškai neįgaliesiems jau šiandien įvairiose pasaulio šalyse naudojama simbolių kalba. Įvairioje užsienio specialiojoje literatūroje šiai kalbai apibūdinti naudojami terminai - Augmentatyvioji ar Alternatyvioji Komunikacija (AAK) arba pagalbinė komunikacija. Ji yra iš esmės nauja bendravimo forma, todėl daugiau teorinė - praktinių žinių, o tuo labiau, specialių techninių priemonių turime labai mažai.

AAK - tai įvairių simbolių kalba. Ji naudojama tada, kai neįgalusis turi kalbėjimo problemų, t.y. negali komunikuoti garsine kalba, neįgalus patenkinti savo gyvenimiškų poreikių, suprasti vienas kito. Įvairios AAK rūšys pritaikomos ir patenkina neįgaliųjų poreikį kalbėti, skaityti ar net rašyti (perduoti ženklus popieriumi). AAK naudojama visur - ugdymo įstaigose, ligoninėse, aptarnavimo sferoje, įvairiose įstaigose, net aukštosiose mokyklose.

Yra įvairūs AAK lygiai, kurie pasirinktinai pritaikomi, atsižvelgiant į intelektą:

- bendravimas konkrečiais daiktais;
- bendravimas nuotraukomis;
- bendravimas paveikslais;
- bendravimas piktogramomis (stilizuoti paveikslai);
- bendravimas BLISS simboliais ( tai tarptautinė simbolių kalba, turinti labai plačias naudojimo galimybes, populiariausi iš jų, tai stilizuoti linijiniai, smarkiai supaprastinti piešiniai).

Įvairiose pasaulio šalyse (Kanada, JAV, D. Britanija, Skandinavijos šalyse) yra naudojamos skirtingos AAK sistemos ir metodikos.

Negalią turinčių žmonių gyvenime vis didesnę vaidmenį ima vaidinti kompiuteriai ir specialioji programinė įranga. Jie padeda lavinti įvairius komunikacijos pojūčius (į ką atkreipti dėmesį, kas ne taip svarbu ir t.t.) ir moko geriau orientotis aplinkoje, lavina negalią turinčiųjų vaikų savarankiškumą.

Taigi šio projekto pagrindiniai tikslai ir išsikelti uždaviniai ir yra sumažinti susidariusį didelį kalbos barjerą tarp neįgaliųjų ir sveikų žmonių, panaudojant jau senai žinomus ir naudojamus neįgaliųjų bendravimo metodus. Bandoma pritaikyti šiuolaikinę kompiuterinę techniką bei išnaudoti šiuolaikinių informacinių sistemų pritaikomumą realioje situacijoje. Kiek galima daugiau išaiškinti apie su tokiu specifinius bendravimo sunkumus turinčius neįgaliuosius ir suteikti kaip galima patogesnius bendravimo su išoriniu pasauliu įnagius.

Šio darbo metu buvo suprojektuota ir realizuota sistema palengvinanti neįgaliesiems skirtų bendravimo metodų panaudojimą. Pagrindinis sistemos sukūrimo tikslas ir buvo suteikti pagalbą bendraujant su fiziškai ir protišcai neįgaliais vaikais ir jų tėvais ar auklėtojais, pritaikant specialius tokio bendravimo metodus.

Sistemos veikimas analogiškas realiai sukurtiems Norvegijos daktarų bendravimo kortelėmis metodams. Pagrindinis skirtumas tarp egzistuojančių metodų ir sukurtos sistemos tai, jog sukurtoji sistema naudoja modernias informacines technologijas ir naujus sprendimus problemos sprendimui įgyvendinti. Iki šiol naudojami metodai paremti dideliu popierinių kortelių kiekiu (apie 390 kortelių) kurį kiekvienas fiziškai (ir dalinai protišcai) neįgalusis vaikas turi įsisavinti, o vėliau ir pasinaudoti norėdamas išreikšti savo veiksmus, emocijas ir būsenas. Sukurtoji sistema leidžia sumažinti krūvį šiems vaikams ir susitvarkyti su kortelių kiekiu lengviau ir paprasčiau, be to programa nereikalauja papildomų priedų vartotojo sistemoje.

Visos popierinės kortelės darbo metu buvo perkeltos į sistemą dinamišku pavidalu ir vėliau panaudotos elektroniniu formatu. Pagrindinės varančios priežastys kodėl pasirinktas toks sprendimo būdas tai netradicinis informacinių technologijų taikymas, darbo naujumas ir tai, jog vaikai labiau įsisavina naudingas žinias naudodami-žaisdami kompiuteriu negu naudodami popierines ideogramas.

Bendraujant su vaikų abilitacijos centru buvo surinkta daug naudingos informacijos apie vaikus ir jų auklėtojas kurios nuolat dirba su šiais vaikais ir žino kokios pagrindinės kylančios problemos naudojant šią kortelių metodiką.

Šios sistemos pritaikymo galimybės yra dar smarkiau išnaudojamos, nes sistemos veikimo principas pagrįstas internetiniu protokolu bei nauja AJAX technologija. Sujungus jas abi buvo

pastebėtas darbe atsiradęs lankstumas ir naudingumas. Interneto protokolas leidžia sistemai veikti universalesniu pritaikymo spektru panaudojant sistemą įvairesnėje aplinkoje.

Darbe nagrinėjami tokio tipo problemoms spręsti, pasaulyje siūlomi, analogai kurie suformuluoja tikslesnius konkretaus darbo siekius ir apibrėžia akivaizdaus funkcionalumo reikalingumą.



## 1. Sistemos paskirtis

### 1.1. Sistemos kūrimo pagrindas

Šis projektas pradėtas norint įgyvendinti galimybę ir neįgaliesiems žmonėms bendrauti, išnaudojant kompiuterines priemones.

Apsilankius KVAC ir susipažinus su naudojamais mokymo-bendravimo metodais ir naudojamomis darbo priemonėmis su fiziškai neįgaliaisiais vaikais buvo įvertinta, kad tokios pat mokymo sistemos kompiuterizavimas t.y. IS sukūrimas, duotų efektyvesnių rezultatų nei dabar naudojami metodai. Pagrindiniai IS privalumai prieš tradicinius būdus, būtų jos universalumas, paprastumas ir neįgaliųjų vaikų motyvacijos bendrauti skatinimas (KVAC duomenimis vaikams labai patinka kompiuteriai ir kompiuteriniai žaidimai).

Kuriant projektą buvo remtasi kriterijais, jog ši kompiuterinės programinės ir aparatūrinės įrangos niša nėra užpildyta, arba tokios programinės įrangos yra labai mažai ir jos tendencija didėti labai maža. Ypač mažai yra tokio tipo lietuviškos programinės įrangos, su lietuviškais dialogais, langais, pranešimais ir programos interfeisu. Taigi kitas šios sistemos kūrimo pagrindimas atsirado kai buvo pastebėtas tokių sistemų trūkumas ne tik Lietuvoje, bet ir užsienio šalyse. Lietuvoje žengiami tik pirmieji tokių autistiškų vaikų gydymo žingsniai ir pritaikomi pasaulyje naudojami gydymo metodai, kurie naudojami pakankamai seniai.

Sukurtas produktas pagelbės daugeliui neįgalių vaikų, taip pat jų tėvams, jų mokytojoms, jų artimiesiems lengviau bendrauti ir suprasti vienas kitą įveikti juos ribojantį kalbos barjerą. Savo projektu tikimės pagelbėti ne vienam Lietuvos vaikui turinčiam fizinę negalę, kuri jam tiesiogiai trukdo atlikti vieną iš pagrindinių žmogaus prigimtinių funkcijų – bendravimą su jį supančiu išoriniu pasauliu, bei taip mokinti suprasti jį supančią aplinką.

### 1.2. Sistemos tikslai

- Sukurti bendravimo sistemą kompiuteriu, internete, adaptuotą neįgaliesiems vaikams, jų mokytojams, tėvams ir jų draugams.
- Pritaikyti informacines technologijas specifinėje srityje.
- Išsiaiškinti neįgalių vaikų poreikius IS
  - išvaizdai,
  - patogumui,
  - funkcionalumui,

- vaizdinės informacijos gausai.
- Sumažinti bendravimo barjerą tarp neįgaliųjų ir juos supančių aplinkinių žmonių.

### *1.3. Organizacijos veiklos analizė*

Mēs bendradarbiaujame kartu su Kauno vaikų abilitacijos centru. Tai specialus neįgalių vaikų darželis – mokykla kuriame šie vaikai specialiais lavinamaisiais žaidimais mokomi tobulinti savo įgūdžius, bendrauti su bendraamžiais. Svarbiausia šitokio ugdymo esmė elgesio formavimas per pagalbos suteikimą ir jos nutraukimą, skatinant sėkmingą tarpusavio suartėjimą. Sulaukus teisingo atsakymo nedelsiant vaikas skatinamas. Skatinimo pavyzdžiai: maistas/gėrimas, žaidimas su mėgstamu žaislu, mėgstamos knygos vartymas bei socialinis atlygis (pagyrimas).

Grupėje su vaikais individualiai dirbama 15-20 valandų per savaitę pagal ABA (taikomojo elgesio analizės) intensyvios terapijos programą (kai kuriuose šaltiniuose dar minima kaip „Lovaas“ metodas).

Turintieji raidos sutrikimų gali sėkmingai mokytis pagal bendrąją ugdymo programą, bet savarankiškai nesugeba nueiti į mokyklą

Lietuvoje maždaug 10-12 žmonių iš 10 tūkstančių gyventojų diagnozuojami autizmo sukelti raidos sutrikimai. Autizmas lemia nepakankamą vaiko socialinių įgūdžių, kalbos, elgesio išsivystymą ar šio vystymosi sulėtėjimą.

Ugdant autistišką vaiką dedamos visos pastangos, kad būtų sukurtos tokios ugdymo sąlygos, kuriose būtų kaip galima padidinti vaiko sėkmę ir sumažinti jo nesėkmes. Reikalavimai supaprastinami taip, kad jie pagelbėtų vaikui atsakyti teisingai ir paremtų jo tinkamą elgesį. Taip pat stengiamasi garantuoti vaiko, dalyvaujančio ugdymo procese "motyvaciją".

Labai svarbu pradiniam etape ugdyti kartojant veiksmus arba užduotis – tai ypatinga biheivioristinė seka reikalinga ugdymo efektyvumui didinti. Šią seką sudaro trijų dalių ugdymo vienetai (I – R – P) (instrukcija – reakcija – paskatinimas), kiekvienas iš jų remiasi kartojimu.

Kartojant veiksmus ar užduotis vaikas išmoksta arba veiksmai/užduotys įtvirtinami. Taip pat šis ugdymas padeda pedagogui išlikti nuosekliam ir lengviau įvertinti vaiko pažangą:

1. Instrukcijos supratimų įtvirtinimas - instrukcija (I). Instrukcija turi skambėti aiškiai, paprastai ir garsiai. Pradiniam ugdymo etape svarbiausia, kad vaikas sukauptų dėmesį instrukcijos išklausymui. Negalima instrukcijos kartoti be rezultatų (arba jei nėra atsakymo). Instrukcija pradžioje turi būti įsakmi ir skambėti garsiau nei kasdienė kalba, vėliau tariama natūraliu garsumu.

2. Atliekamo veiksmo kartojimas - reakcija (R). Jei nėra jokio atsakymo, tai laikoma neteisingu atsakymu. Taikomi nuoseklūs kriterijai. Reikia įsitikinti, ar vaikas elgiasi teisingai. Jei vaikas neatsako, sufleruojame teisingą atsakymą, jei neatlieka veiksmo tada iš karto suteikiama fizine pagalba.

3. Paskatinimų kartojimas - paskatinimas (P). Vaikas už teisingą atsakymą nedelsiant apdovanojamas. Tam naudojami įvairūs atlygio būdai: gardumynai, gėrimai, mėgiama veikla, socialinis atlygis ir t.t. Darbo patirtis parodė, kad kuo daugiau paskatinimų, tuo pasiekiamas didesnis efektas.

Vienas iš autistiškų vaikų ugdymo metodų yra sufleravimas/pagalba. Sufleravimas gali būti verbalus arba neverbalus. Verbalinis būdas - tai informacijos perdavimas žodžiu ar raštu. Neverbalinis būdas - tai informacijos perdavimas kūno judesiais, nenaudojant žodžių.

Verbalinės bendravimo formos yra dvi:

- Informacija perduodama žodžiu.
- Informacija perduodama raštu.

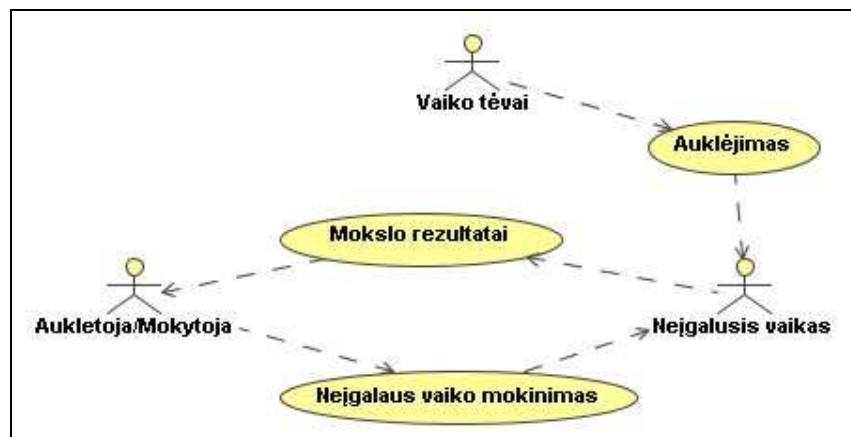
Neverbalinės bendravimo formos išreiškiamos:

- Ženklų kalba, raštu;
- Kūno kalba. Nervingi judesiai parodo nuobodulį, ant klubų sudėtos rankos - pyktį, pečių traukymas – abejojimą. Dažnas burnos srities lietimasis pirštais, pieštuku ar bet kuo kitu, lūpų kandžiojimas ar net žandikaulio lietimasis ranka rodo, kad žmogus nerimauja, kad ta situacija, kuri sukėlė jam poreikį daryti kažkuriuos iš prieš tai išvardintų veiksmų, jam yra stresinė;
- Gestais;
- Balso intonacija;
- Drabužiais;
- Elgesiu;
- Mimika (veido išraiška). Piktas žvilgsnis išreiškia nepasitenkinimą; šypsena - draugystę, laimę, meilę; pakeltais antakiais - abejojimą, nustebimą, siauromis akimis ir suspaustomis lūpomis - pyktį;
- Žvilgsniu. Akies mirksnis parodo susidomėjimą žmogumi, gundymą, viliojimą;
- Prisilietimu;
- Šoku;

- Plojimais.

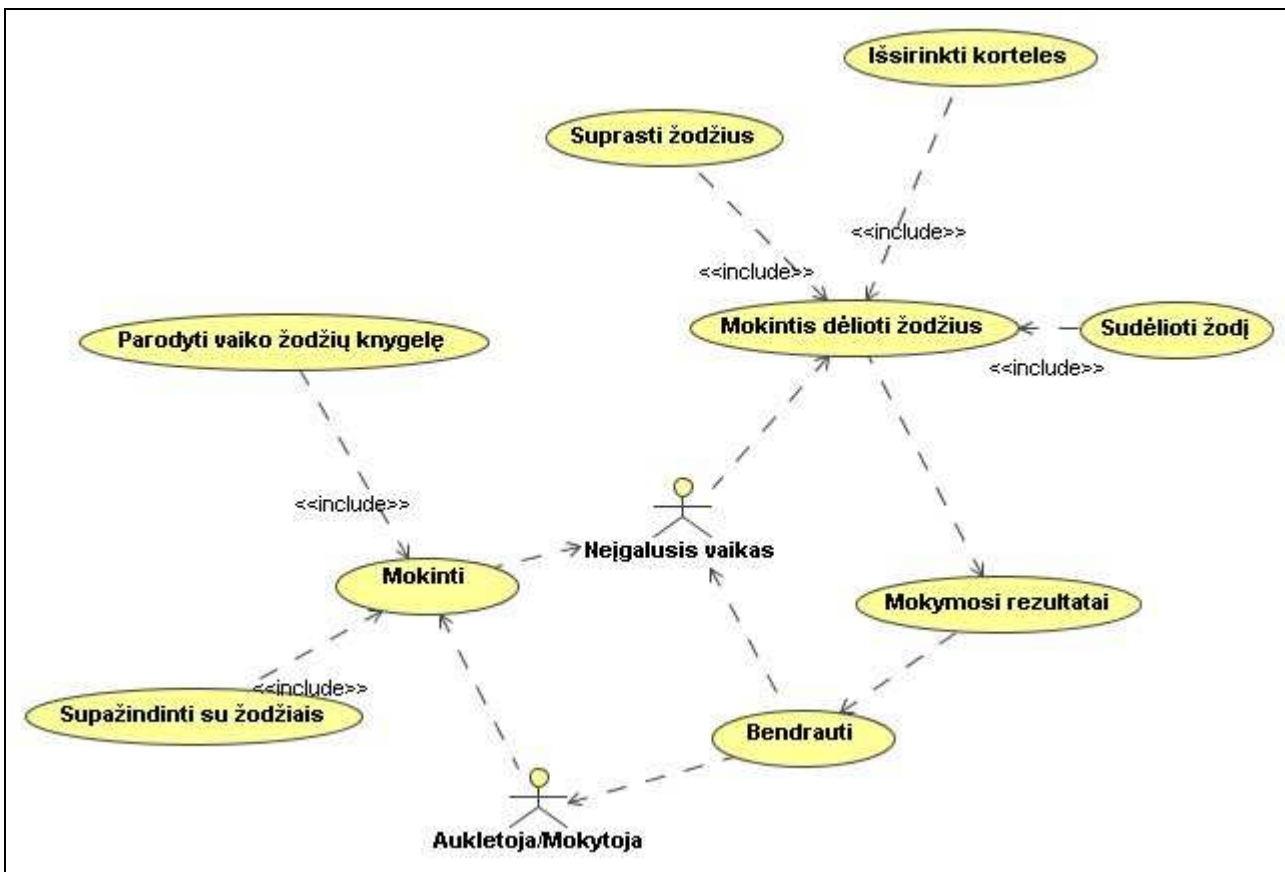
Taigi sufleravimą pateikiame iškart po instrukcijos (per tris sekundes). Jį reikia nutraukti kuo greičiau. Taip pat reikia vengti neapdairių pasakinėjimų – kaip žvilgčiojimo į reikiamą daiktą ar teisingo atsakymo „konstravimo“. Po kiekvieno kartojimo daroma pauzė (viena/trys sekundės), pakankamai ilga, kad būtų galima kartojamus dalykus atskirti tarpusavyje, bet ne per ilga, kad vaikas neprarastų dėmesio ir nepradėtų elgtis kitaip nei iš jo tikimasi. Supažindinant su nauju daiktu/veiksmu kartojama ta pati instrukcija keletą kartų iš eilės. Įsisavintų kartojimų kriterijus: teisingų atsakymų skaičius - 90% t.y. 9 teisingi atsakymai iš 10. Visi parinkti simboliai yra minčių, jausmų ir sąvokų abstrakcijos. Mes turime teisę pasirinkti patį efektyviausią bendravimui būdą. Tačiau negalime pamiršti, kad mes naudojame žodžius - abstrakcijas, todėl kiti šiuos žodžius gali suprasti klaidinga reikšme. Bendraujant reikia vengti verstų iš kitų kalbų žodžių (t. y. vertalų) ir semantizmų.

Kauno neįgaliųjų vaikų abilitacijos centre specialistų komanda pradinio ugdymo programoje išskiria tris pagrindines darbo kryptis: dėmesingumo skatinimą, neverbalinės imitacijos mokymą ir gebėjimo lyginti ugdymą. Aukščiausio lygio organizacijos veiklos analizė pateikiama žemiau (1 pav.)



1 pav. Aukščiausio lygio organizacijos veiklos analizė

Čia pateikiama organizacijos veiklos analizė atspindi tarpusavio sąryšį tarp neįgaliųjų vaikų, auklėtojos/mokytojos ir vaiko tėvų. Kadangi projektas orientuotas į mokymą ir vaikus bus analizuojami Auklėtojos/Mokytojos darbo ryšiai su vaikais. Detalesnio lygio organizacijos veiklos analizė pateikiama žemiau (2 pav.).



2 pav. Detalesnė organizacijos veiklos analizė (use-case)

Iš pradžių vaikai dirbti mokomi duodant jiems kokį nors daiktą ar skanėstą, o vėliau pakanka pagyrimo. Taip vaikai išmokomi suprasti reiškinį priežastis, eigą, padarinius, ko paprastai autizmu sergantys žmonės nesuvokia[16].

### 1.4. Tyrimo sritis, objektas ir problema

Šis projektas orientuotas į fiziškai neįgalius vaikus kurių protiniai sugebėjimai ir intelektas yra išsivystę pilnai arba lyginant su kitais tokio tipo vaikais išsivystęs gana aukštai. Tačiau daugiausia dėl įgimtų neišsivysčiusių fizinių gebėjimų jiems bendrauti su aplinka vis dar pakankamai sunku [18]. Visame pasaulyje nuo seno buvo ir yra kuriami būdai, metodai, pritaikomi pagalbiniai įrankiai, sistemos, kad šiems vaikams palengvinti ir supaprastinti bendravimo su išoriniu pasauliu galimybes ir būdus.

Prieš kuriant sistemą ir prieš įsitikinant tokios sistemos naudingumu, buvo bendrauta su Kauno vaikų abilitacijos centru kurie nuolat suteikė naudingos informacijos apie pas juos besimokančius ir besigydančius neįgaliuosius vaikus.

Šiame centre bendravimui su kai kuriais vaikais naudojamos ideogramos(*idea* – sąvoka, *gramma*- rašytinis ženklas, lot. *pictus* - pieštas). Tokie vaikai su sergantys autizmu būna vidutiniškai 10-12 iš 1000-2000 žmonių. Šios kortelės ir technologija plačiai taikomi ir kartu naudojamas specialus logografinis raštas t.y. kai visos piktogramos išdėstomos nuosekliai ir logiškai iš ženklų derinio sudaromi ne paveikslai, reiškiantys bendrą mintį, o simboliai reiškiantys konkrečius žodžius, sąvokas.

Ideo-piktogramos yra simbolių kalba. Jos parodo daiktus, veiksmus. Šie simboliai dar skirstomi į etnografinius, piktografinius, ideografinius (egiptiečių raštas), tarptautinius ir laisvuosius (dar vadinama Bliss sistema)

Magistriniame darbe naudojamos Norvegijoje sukurtomis mokymo ir bendravimo priemonėmis. Jos skirtos žemo intelekto žmonėms arba vaikams. Tai specialios, juodai baltos kortelės su ant jų išpieštais baltais, daiktais, veiksmais, būdvardžiais, objektais, gyvūnais ir kitais objektais. Viršutinėje šių kortelių dalyje užrašyta kiekvienos kortelės reikšmė. Kortelės pavyzdys pateiktas žemiau 3 pav.



3 pav. Kortelės pavyzdys „patenkintas“

Jos naudojamos kartu su kiekvienam vaikui skirtu nuosavu albumu. Vaikas norėdamas aplinkiniams ką nors pasakyti savo tik jam skirtame albume turi susirasti, susirinkti ir sudėlioti korteles tokia eilės tvarka kokia jis nori pasakyti savo žodelius. Didžiausi žodeliai susideda iš apytiksliai 4,5 kortelių. Šis metodas greitai išpopuliarėjo daugelyje pasaulio šalių savo panaudojimo paprastumu ir nesudėtinga naudojimosi instrukcija, bei efektyviais darbo rezultatais.

Pagrindinis mano magistrinio darbo tikslas ištirti tokių vaikų poreikius ir tikslus, norus (išvaizdai, patogumui, funkcionalumui, vaizdinės ir tekstinės informacijos gausai), siekius ir sukurti būtent jiems adaptuotą kompiuterinę, informacinę, mokomąją sistemą-žaidimą. Jos pagalba neįgalusis vaikas galėtų aplinkiniams pranešti apie savo norus, išgyvenimus ir kitas (artimiausiu metu busimas, esančias ar jau praėjusias) su juo susijusias emocijas. Taip pat išsiaiškinti ar tokio tipo negalią turintiems žmonėms(vaikams) galima pritaikyti kompiuterinę sistemą, ar jam bus lengva joje orientotis, ar jis galės adaptuotis ir ar jiems bus fiziškai įmanoma (turintiems judesio ir padėties sutrikimų) tokia sistema naudotis. Taip pat bus bandoma pritaikyti ir sistemoje panaudoti

kompiuterinius, specialiai neįgaliesiems skirtus, įvedimo įrenginius lengvinančius bendrą sistemos veikimo procesą.

Šiuo magistriniu darbu ir jame surinkta informacija, tikimės padėti ne tik neįgaliesiems žmonėms, bet ir ateityje atsiradusiems panašių sistemų kūrėjams dar labiau tobulinti, vystyti ir plėsti darbus čia aprašoma tema.

Tuo tarpu magistrinio darbo metu sukurta sistema tikimės, kad jis padės lavinti, vystyti ir tobulintų šių neįgalių vaikų bendravimo įgūdžius, su aplinkiniais, skatinti ir lavinti protinę veiklą, sumažinti iki minimumo naudojimosi sistema sunkumą, pritaikyti svetainės interfeisą šiems žmonėms priimtinomis spalvomis, raidėmis, simboliais ir kitais žymėjimais.

Galiausiai pasiekti kuo tikslesnius, pateiktai užduočiai, rezultatus, juos apibendrinti, padaryti išvadas, atlikti statistinius ir kitus reikalingus skaičiavimus kurie galėtų vėliau padėti tobulinti projektą naujomis idėjomis. Projektuojama sistema taip, kad ateityje būtų galimas jos prijungimas prie interneto ar vietinio tinklo. Tai daroma norint išplėsti šį projektą ir suteikti galimybę šiems žmonėms bendrauti tarpusavyje ženklų bei simbolių pagalba, bei stengtis sukurti ir suprojektuoti komunikacijos tarp kelių neįgalių žmonių tinkle funkcijas ir metodus. Atlikti ir ištirti panašių sistemų veikimo būdus, pateikti analizę. Sukurtą sistemą patalpinti viešai prieinama - internete.

### *1.5. Išvados*

- Šiame skyriuje aptartos išskylančios bendravimo problemos neįgaliesiems su juos supančiais aplinkiniais ir su savo bendraamžiais. Apžvelgta magistrinio darbo tyrimo sritis, numatomi tikslai ir IS kūrimo pagrindimas.
- Išsiaiškinti bendravimo metodai darbui su autizmu sergančiais neįgaliaisiais vaikais. Aptarta organizacijos veiklos analizė bei darbo metodai.
- Aprašyta tolesnio darbo tyrimo sritis bei objektas ir problema. Magistrinio darbo metu kuriama sistema yra vienintelis tokio pobūdžio produktas, pritaikytas neįgaliesiems vaikams ir naudojamas kaip žaidybinė, mokomoji, bendravimui skirta priemonė.

## 2. Sistemos analizė

### 2.1. Analoginės sistemos

Darbe pateikiamos programos ir programinė įranga skirta neįgaliesiems. Programos dalinai panašios į kuriamą sistemą, bet kartu ir skirtingos, tačiau jas galima įvardinti tik kaip analogines sistemas. 1 lentelėje išdėstomi kiekvienos programos operacinė sistema, aprašymas privalumai ir trūkumai. Visos aptartos programos rastos internete ir išbandytos demonstracinės jų versijos kurios dalinai atspindi veikimo principus ir 1 lentelėje aprašomas savybes.

**1 lentelė. Analoginės sistemos**

<i>Produkto pavadinimas</i>	<i>Operacinė sistema</i>	<i>Programos aprašymas</i>	<i>Privalumai (lyginant su kuriama sistema)</i>	<i>Trūkumai (lyginant su kuriama sistema)</i>
AccessMaths 4.2	Windows	Tai programinė įranga skirta piešiant pieštuku ir užrašinėjant skaičius bei simbolius spręsti matematinės užduotis ir lygtis. Specialiai pritaikytas vaikams su fizine negale.	<ul style="list-style-type: none"> <li>- Naudingas sprendimas</li> <li>- Gera idėja</li> <li>- Grafikos panaudojimas</li> <li>- Dažniausiai naudojamų įrankių išsirinkimas</li> <li>- Palaiko šablonus sukurtoms matematinėms išraiškoms</li> </ul>	<ul style="list-style-type: none"> <li>- sunkus valdymas</li> <li>- sunkiai suprantama aplinka ir jos veikimas</li> <li>- Mokama (30 dienų galima naudotis nemokamai)</li> <li>- Reikia gerai žinoti anglų kalbą (nepalaikomas daugiakalbiškumas)</li> </ul>
HotSpots 2.0	Windows	Programinė įranga skirta specialiai neįgaliesiems kurie negali naudotis kompiuterio pele o vietoje to gali naudoti liečiamąjį ekraną. Programos pagalba galima sudėti „karštus taškus“ kurie paspaudžia pelės klavišą vos jie priliečiami ant ekrano.	<ul style="list-style-type: none"> <li>- minimalus interfeisas</li> <li>- nemokama</li> <li>- tiesioginis palaikymas</li> </ul>	<ul style="list-style-type: none"> <li>- sunku nusistatyti taškus be pagalbos</li> <li>- sudėjus daug „karštų“ taškų nebelieka vietos vizualiam vaizdai</li> <li>- Angų kalba</li> </ul>



Number Navigator 2.0	Windows	Žaidybinė mokomoji programa, emuliuojanti matematikos langeliais pieštą sąsiuvinį. Langeliuose rašomi skaičiai ir pagrindiniai matematiniai simboliai sąveikauja (skaičiuojami) tarpusavyje žemiau pateikdami gautus rezultatus.	<ul style="list-style-type: none"> <li>- Gera mintis</li> <li>- Suprantamas interfeisas</li> <li>- Draugiška vartotojo aplinka</li> <li>- Nemokama nekomerciniam naudojimui.</li> </ul>	<ul style="list-style-type: none"> <li>- Anglų kalba</li> <li>- Nedaug funkcijų palengvinančių naudojamąsi programa</li> </ul>
Prophet Multi 1.0	Windows	Programa skirta palengvinti žodžių rašymą kompiuteriu tekstiname Microsoft Word dokumente. Vos pradedamos rašyti pirmos žodžio raidės programa suveikia ir pateikia galimus tolesnius žodžio variantus vartotojui belieka pasirinkti žodį kuris susietas su kompiuterio klaviatūros klavišais.	<ul style="list-style-type: none"> <li>- Darbas tik klaviatūra</li> <li>- daugiau nei 10000 žodžių</li> <li>- yra žodžių reitingavimas</li> <li>- įdėtas greitasis vedlys</li> <li>- 30 dienų veikianti versija</li> </ul>	<ul style="list-style-type: none"> <li>- sukurtas tik anglų kalbos žodynas</li> <li>- nepalaiko Windows XP</li> <li>- mokama</li> </ul>
SAW 4.5	Windows	Pagrindinis programos šūkis „Special Access to Windows“ (Specialus Windows naudojimas). Tai programa palengvinanti darbą žmonies kuriems sunku panaudoti pelę. Tai liečiamajam ekranui ar kitam panašiam įrenginiui pritaikyta programa ekrane pateikianti dauguma raidžių ir kontrolinių simbolių. Programos kūrėjai ateityje žada atnaujinti iki piešimo galimybių.	<ul style="list-style-type: none"> <li>- tinka daugeliui taikomųjų programų</li> <li>- mažos raidės (sunku pataikyti)</li> <li>- palyginus draugiškas vartotojo interfeisas</li> </ul>	<ul style="list-style-type: none"> <li>- sunku dirbti keliais langais</li> <li>- ne visada simboliai siunčiami ant aktyvaus lango</li> <li>- mokama</li> <li>- Angliškas (lotynų raidės) žodynas</li> </ul>
Boardmaker & Speaking Dynamically Pro Combo	Windows	Tai mokymo įrankis savyje palaikantis 3000 įvairių simbolių, kuriuos galima dėlioti, rūšiuoti ir išsispausdinti kaip mokomąją priemonę neįgaliesiems vaikams.	<ul style="list-style-type: none"> <li>- Spalvoti simboliai</li> <li>- Galima spausdinti įvairiais dydžiais</li> <li>- Galima piešti savo</li> </ul>	<ul style="list-style-type: none"> <li>- Skirta tik mokytojams</li> <li>- Mokama</li> <li>- Ne visi paveikslukai tinka spausdinimui</li> </ul>

The Language of Bliss	Mac/ Windows	Tai „Bliss“ simboliai skirti vaikams, šiais simboliais išreikšti veiksmi, jausmai, emocijos, daiktai kiti objektai ir detalės. Kiekviena kortelė pažymėta simboliu ir turi savo pavadinimą	- Daugybė simbolių - Galima išreikšti beveik visas emocijas ir norus	- Labai sudėtingas tiek vaikams tiek mokytojams - Mokamas - Anglų kalba
Letter Match! Alphabet Memory Game	Windows	Programa skirta autistiškam vaikui mokintis atpažinti abėcėlę.	- Gražus interfeisas - Didelės raidės	- Mažas funkcionalumas - Anglų kalba - Programa mokama - Tik pele valdoma sistema
"Hot! Hot! Hot! " Fire Safety	Windows	Programos paskirtis per žaidimus mokinti vaikus apie karštų daiktų suvokimą ir apsaugą nuo jų. Viskas iliustruota ir vaizdžiai išdėstoma	- Mokoma per žaidimus	- Mokama - Anglų kalba - Neišnaudotas ekranas
Fire & Street Safety Interactive Program	Windows	Programa skirta kaip informacinė priemonė ką gali padaryti, kad neįvyktų gaisras ir kokios nelaimės gali atsitikti gatvėje ir kaip jų išvengt, ką daryti kai pamatai nelaimę gatvėje ir kita. Viskas iliustruota ir įgarsinta dainelėmis.	- Patraukli - Lengva ir paprasta naudoti - Animuota	-Mokama -Anglų kalba - Pritaikyti ženklai tik kaire puse važiuojančiam transportui

Čia aprašomi privalumai ir trūkumai į kuriuos bus atsižvelgta projektuojant ir kuriant savo sistemą. Privalumai kuriais pasižymi kuriamoji sistema:

- Lietuvių kalba,
- Paprasta naudojimo aplinka,
- Palaiko visas operacines sistemas kur veikia grafinės interneto naršyklės(IE6+, FF1.5+),
- Prieinamumas internetu,
- Nereikia instaliuoti,
- Nenaudoja disko resursų,
- Nemokama,
- Kortelių reitingų sistema,
- Bendravimo palaikymas.

Šis funkcionalumas buvo išrinktas iš analoginių programų privalumų ir panaudotas kuriamoje sistemoje tikintis, kad tai įtakos geresnį vartotojų prisitaikymą prie interfeiso ir greitesnį funkcionalumo supratimą, bei paskatins norą bendrauti.

## *2.2. Analizės metodų, priemonių parinkimas*

Analizė buvo atliekama naudojantis informacinėmis ir mokomosiomis knygomis apie neįgaliuosius, įvairiais internetiniais šaltiniais ir papildoma informacija suteikta Kauno vaikų abilitacijos centre.

Pagrindiniai analizės metodai:

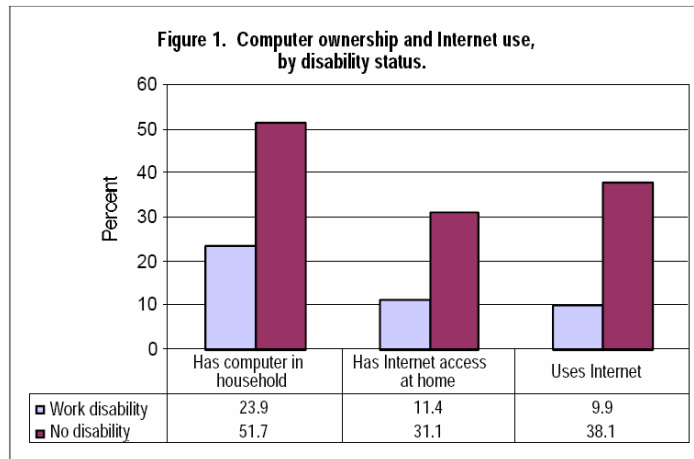
- Išsiaiškinti neįgaliųjų poreikius – šios analizės metu, bandoma kuo nuodugniau ir smulkiau sužinoti apie sistema besinaudojančių žmonių būklę, aplinką, mėgiamus ir nemėgiamus objektus, žymėjimus, spalvas, animaciją, garsus, norą dirbti su tokia sistema, palankiausių objektų atvaizdavimo ir išdėstymų ekrane būdus ir t.t..
- Išsiaiškinti vartotojus – sužinoti ko vartotojas gali, ko negali, ką sugeba įsiminti ir ko nesugeba, išsiaiškinti bendravimo su kompiuterine sistema supratimo lygį, kokie kompiuteriu atliekami vartotojui veiksmai patinka labiausiai ir kas nepatinka, vargina, ir ar užtenka įgūdžių, fizinės galios, koordinacijos. Taip pat sužinoti galimas vartotojų grupės vyraujančias amžiaus ribas.
- Išsiaiškinti tokių sistemų veikimo principus – paanalizuoti kaip panašias sistemas kuria kitos organizacijos, kitos šalys, kaip jiems sekasi, sužinoti naudingų patarimų ir pasiūlymų, pasisemti naujų idėjų ir didesnės, tokios neįgaliesiems skirtos sistemos, programavimo patirties.
- Atlikti panašių sistemų analizę – ši analizė leis išsiaiškinti panašių sistemų privalumus ir trūkumus, tokių sistemų gausą, naudojimosi jomis populiarumą. Analizės metu aprašysim detalų tokių sistemų veikimą, principus, tipą, kainą. Palyginsim su kuriama sistema, interfeiso, patogumo, ir naudingumo požiūriu. Sužinosime kokio funkcionalumo dar trūksta, kas atlikta ne taip, arba ką galima pataisyti geriau. Vėliau visus atrastus naujus privalumus pabandydysime pritaikyti savo kuriamoje sistemoje ir naudojamoje metodikoje.

- Sužinoti pritaikymo būdus – tai analizė kurios metu išsiaiškinsime kur galėsime panaudoti savo sukurtą sistemą, sukurtus metodus ir priemones ar jie naudojami ten kur įmanoma juos naudoti, gal yra kitokie panaudojimo atvejai kuriems tinka sukurta sistema.
- Išsiaiškinti darbo priemones – analizė turėtų pagelbėti išsirinkti projektuojamai sistemai projektavimo, programavimo, priežiūros, duomenų saugojimo ir skaitymo įrankius, greičiausiai būdus, mažiausiai resursų ir naudojančias aplikacijas, lengviausius būdus pasiekti tikslą. Atkreipti dėmesį į vartotoją, aplinką kurioje bus dirbama, tinklo protokolus kuriais bus perduodami duomenys, užtikrinant sistemos saugumą ir užkirsti įsilaužimo, privatumo pažeidimo galimybes.
- Apsibrėži sisteminius reikalavimus – kokioje sistemoje šis projektas bus naudojamas, kokie resursai, servais, operacinė sistema, programinė įranga, aparatūrinė įranga - reikalingi. Kaip efektyviai išnaudoti resursus, naudojantis grafika, ir nepasinaudojant primityviomis programavimo kalbomis.

## *2.3. Neįgaliųjų poreikiai sistemai*

### *2.3.1. Vartotojo sąsaja*

Neįgaliųjų varotojo grafinė sąsaja turi didelę reikšmę neįgaliesiems. Vien tai, kad jiems tenka įveikti savo negalę jau smarkiai riboja naudojimąsi kompiuteriu ir kompiuterinėmis sistemomis, o neradus patrauklaus varianto ir nuvilia neįgalųjį vartotoją ir taip pranyksta paskata nepatogia sistema naudotis. Taigi pastangos padaryti kažką unikalios ir paprastesnės naudoti priklauso nuo to kaip bus pažiūrėta pačioje sistemos kūrimo pradžioje. Kokie aspektai ir išoriniai veiksniai labiausiai įtakos sistemos sąsajos kūrimo kryptį. Dauguma neįgaliųjų neturi kompiuterio, o dar didesnė dalis ir interneto. Amerikos neįgaliųjų vartotojų procentas turintis internetą pavaizduotas žemiau (4 pav.).



**4 pav. Kompiuterį turinčių neįgalių ir sveikų žmonių skaičius Amerikoje.**

Čia šviesesne(kairiau) grafiko spalva pažymėti žmonės su vienokio ar kitokio tipo negale, o tamsesne(dešiniau) spalva pažymėti sveiki žmonės. Vertikalėje nurodyti procentiniai įverčiai, o horizontalėje paėiliui „turi kompiuterį namuose“, „turi interneto prieigą“, „naudojasi internetu“. Kaip matyti iš grafiko tai didžiulė problema ne tik Lietuvos mastu bet ir kitur pasaulyje.

Vartotojo valdymas turi būti suprantamas, kiek galima paprastas ir intuityvus. Valdymas turi būti realizuotas bent jau standartinė kompiuterio pele ir klaviatūra arba pritaikant specialias neįgaliesiems skirtas įvedimo įrenginius. Kiekvienas veiksmas gali būti atliekamas pasirinktinai vien klaviatūra arba vien pele, bet galimas ir abiejų įrenginių valdymo derinys.

Svetainės realizavimas turi būti informacinio – žaidybinio pobūdžio. Svetainės turinys neturi sukelti priešišku jausmų ir nepageidaujamų blogų emocijų. Spalvos sistemoje turi būti parinktos kuo šviesesnės ir kuo neutralesnės – geriausia iš pastelinės gamos derinio. Ekranų spalvinis fonas gali būti ramus, tolygus ir kartu neatitraukiantis vartotojo sutelkto dėmesio nuo jo pagrindinio tikslo šioje svetainėje. Neatsitiktinai šiame projekte bus kuriama tokia sistema kuri būtų aiški, suprantama, paprasta, tiesiog primityvi, bet kartu ir nenusileistų kitiems panašios paskirties produktams savo funkcionalumu. Programos interfeisas pasirinktas specialiomis, su niekuo neasocijuojančiomis, neutraliomis spalvomis ir atspalviais.

Svetainės vartotojui kiekvienas jo veiksmas neturi asocijuotis su sunkiai įvykdoma užduotimi t.y. valdymas turi būti kuo paprastesnis ir kuo lengviau suprantamas šio specifinio tipo vartotojui.




### 2.3.2. Pritaikomos įvedimo sistemos

Sistemai lengvai pritaikomos specialiai neįgaliesiems skirtos įvedimo sistemos. Jos buvo parinktos ir išanalizuotos, bei sugrupuotos pagal įvairius neįgaliųjų poreikius.

Įranga kuri padeda patogiam darbui išdėstyti ir pritaikyti prie vartotojo monitorių, klaviatūrą dokumentaciją ir įrengti darbo vietą yra naudojami daugeliu individų su negalia.

Tačiau kiek sudėtingesnės negalios rūšys reikalauja iš neįgaliojo daugiau pastangų. Kaip padaryti, kad neįgalusis turėtų atlikti kuo mažiau operacijų ir pasiekti kuo tikslesnius norimus rezultatus? Tam naudojama speciali įvedimui skirta aparatūra ir programinė įranga, jaučianti mažiausius rankų pirštų, burnos, liežuvio, galvos bei kitų kūno dalių judesius ir virpesius. Šiuolaikinės operacinės sistemos jau dalinai pritaikytos neįgaliesiems, jau egzistuoja „lipnieji klavišai“ (angl. stickiekeys), „filtravimo klavišai“ (angl. filterkeys) ir „pažymėtieji klavišai“ (angl. togglekeys). Neįgaliesiems skirti valdymo įrenginiai ir jų tipai pateikiami 2 lentelėje.

**2 lentelė. Neįgaliesiems skirti valdymo įrenginiai**

<i>Paveikslėlis</i>	<i>Aprašymas</i>
<p style="text-align: center;"><b>ABC Layout</b></p> 	<p>Klaviatūra su abėcėlės tvarka išdėstytais dideliais pagrindiniais klavišais. Ji patogi tuomet kai vartotojui sunku prisiminti kur yra klavišai ir kai sunku nusitaikyti tiksliai paspausti norimą klavišą.</p>
<p style="text-align: center;"><b>QWERTY Layout</b></p> 	<p>Klaviatūra su „QWERTY“ tvarka išdėstytais dideliais klavišais. Patogi kuomet reikia didelių klavišų ir kai sunku nusitaikyti.</p>
	<p>Klaviatūra su paveikslėliais vietoj standartinių klaviatūros klavišų. Tinka tik specifiniams poreikiams ir tik tam tikroms programoms.</p>

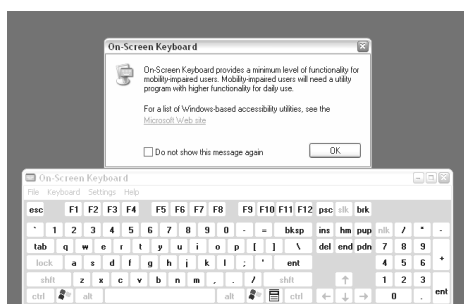


Sukurtos klaviatūros kurioms rankų judėjimas gali būti minimalus. Ši klaviatūra su ergonomiškais ir minimalių rankų judesių reikalaujančiais klavišais.

Ranka įstatoma į įrenginį visais pirštais ir sukiojant juos kiekviena į keturias skirtingas pusės išgaunamos ant piršto keturios skirtingos raidės ir raidžių kombinacijos.



Sukurtos ir specialios vienos rankos klaviatūros. Jos ypač naudingos kai negalima valdyti vienos iš rankų. Tokiu atveju neįgalusis pasinaudodamas tik viena ranka gali sėkmingai naudotis kompiuteriu ir spausdinti visomis raidėmis.



Be standartinės kompiuterinės klaviatūros patogų naudoti ir programiškai emuliuojamą virtualią klaviatūrą. Tokia galimybė taip pat jau egzistuoja naujausiose operacinėse sistemose.



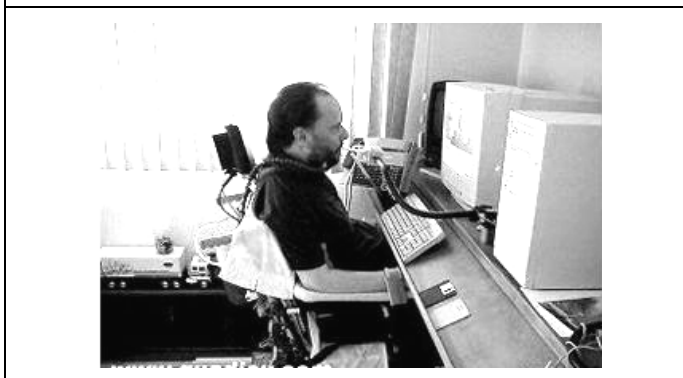
Neįgalieji kurie negali daryti plačių mostų ir turi koordinacijos sutrikimų naudoja mini klaviatūrą. Ji naudinga kai reikia mažais judesiais parašyti daug raidžių. Tam naudojami specialūs padėklai su raidėmis ir prijungtu elektroniniu pieštuku.



Kitokio tipo įvedimo įrenginiai tai kompiuterinės pelės ir jų alternatyvūs variantai neįgaliesiems. Viena iš tokių pelių tai kojomis valdoma kompiuterinė pelė jungiama per PS/2 prievadą. Jai nereikalinga rankų pagalba todėl tai savotiškai pelės valdymo rankomis krūvis perteikiamas kojoms. Kiekvienas tokios pelės padas 360 laipsnių kampu aktyvus paspaudimams ir judesiams kurie ir imituoja pelės judesius.



Kita labai naudinga ir lengvai panaudojama kompiuterinės pelės alternatyva skirta žmonėms galintiems laisvai sukinti galvą ir sunkiai valdantiems rankas bei kojas. Tai Infraraudonųjų spindulių signalu veikianti aparatūra. Viena dalis montuojama prie kompiuterio ekrano kita tvirtinama prie neįgaliojo galvos(daviklis). Savo galvos judesiais žmogus kontroliuoja pelės judesius aukštyn, žemyn, kairėn, dešinėn.




Dar viena pelės alternatyva (angl. k. Joystick). Tai pelė kuri montuojama prie kompiuterio stalo tiesiai prieš žiūrovą. Ji valdoma burnos judesiais, liežuviu ir burna sukiniant šią pelę. Paspaudimai imituojami liežuviu stumtelint valdymo kotelį nuo savęs.



Dar viena pelės alternatyva tai ratukinė pelė. Jos veikimas pagrįstas „apverstos pelės“ principu t.y. jos ratukas perkeltas į viršutinę pelės dalį.



	<p>Pelės būna ne tik kietos ir šaltos. Kita dalis pelių lietimui jautrios pelės. Verta pažymėti, jog šiuolaikiniai nešiojamieji kompiuteriai taip pat jau turi lietimui jautrias peles.</p>
---	---

Šioms klaviatūroms nereikalingos jokios papildomos programos ar kompiuterinės tvarkyklės norint naudotis sukurta sistema, nes jų veikimas iš esmės nesiskiria nuo klaviatūros ar pelės standartinio veikimo. Sistemos kūrimas paremtas mintimi jog vartotojas gali naudoti tik klaviatūrą arba tik pelę, o esant reikalui ir abu įvedimo įrenginius kartu.

## *2.4. Projektavimo metodų, priemonių parinkimas*

Pagrindinis programos valdymas realizuojamas klavišų ir pelės paspaudimais nors daugiausia programoje projektuota objektų kurie valdomi kursoriumi.

Vartotojo įvedimas gali būti atliekamas pasirinktinai:

- Klaviatūra,
- pele,
- liečiant jautrų ekraną,
- specialia neįgaliųjų klaviatūra,
- klavišais valdoma pele.

Darbe parinktos intuityvios ir paprastos projektavimo priemonės. Darbe naudojama UML specifikavimo, modeliavimo ir specifikacijų kūrimo kalba, skirta specifikuoti, atvaizduoti ir konstruoti programos dokumentaciją.

## *2.5. Išvados*

- Analizės metu buvo detalizuotas tyrimo objektas ir aprašyti tyrimo metu iškilę probleminiai klausimai bei probleminės situacijos.
- Buvo surastos pasaulyje sukurtos panašios sistemos. Išrinktos gerosios ir blogosios rastų sistemų savybės. Atrinktos gerosios savybės ir funkcionalumas buvo pritaikytos ir panaudotos šiame darbe kuriamai sistemai.
- Atsižvelgta į neįgaliesiems išskylančias problemas naudojant kompiuterinę programinę ir techninę įrangą. Sudarytas sąrašas tikslų kurie turi būti įgyvendinti šiame darbe.

- Išsiaiškinta, kad projekte bus kuriama tokia sistema kuri būtų aiški, suprantama, paprasta, tiesiog primityvi, bet kartu ir nenusileistų kitiems panašios paskirties produktams savo funkcionalumu. Atsižvelgiant į neįgaliųjų poreikius, programos interfeisas bus kuriamas specialiomis, su niekuo neasocijuojančiomis, neutraliomis spalvomis ir atspalviais.
- Analizės metu vis dėl buvo įsitikinta kuriamos informacinės sistemos reikalingumu ir praktine nauda.

### 3. Reikalavimų informacinei sistemai specifikacija

#### 3.1. Reikalavimų modelis

Pageidaujiamiems rezultatams pasiekti tam tikru sistemos veikimo metu, kiekvienai sistemos daliai privaloma nustatyti ir aprašyti apribojimus, maksimalias bei minimalias galimybes kurias viršijant sistemoje galima klaidų tikimybė arba neplanuotas veikimas. Tokie apribojimai sudaryti bei galioja šiai sistemai. Kiekviena svarbi sistemos dalis prieš kūrimą turi būti pateikta su detalia informacija apie keliamus reikalavimus vartotojams, vartotojų tipus, reikalingas operacines sistemas, be to aprašomi funkciniai ir nefunkciniai reikalavimai keliami sistemai norint realizuoti betarpišką sistemos veikimą.

#### 3.2. Vartotojai

##### 3.2.1. Reikalavimai vartotojams

Projekte vartotojų grupės nėra griežtai apibrėžtos, nes pagrindinis vartotojas į kurį nukreipta ši informacinė sistema yra neįgalusis vaikas ir jo gebėjimus ją suprasti ir palaipsniui įsisavinti bei valdyti. Todėl pagrindinis vartotojas (neįgalusis) yra priklausomas nuo kito vartotojo (šiuo atveju nuo savo mokytojo, auklėtojo) ir jo veiksmų bei jo sugebėjimo apmokyti neįgalųjį vaiką.

Pati sistema iš vartotojo reikalauja tik pagrindinių kompiuterio valdymo įgūdžių, klaviatūros simbolių suvokimo ir/arba koordinuotų pelės judesių bei dalinio supratimo apie programą ir jos darbo esmę. Be šių gabumų sistemoje reikalingas ir Lietuvių kalbos mokėjimas arba kitokios formos supratimas, kuris leistų neįgaliajam atpažinti ir suprasti reikšmę, sistemos jam pateikiamose simboliuose ir ideogramose, meniu formose languose.

Kiekvienas naujas vartotojas norėdamasis naudotis sistema, turi turėti priėjimą su savo identifikaciniu vartotojo vardu. Tam reikalinga vartotojų registracija šiame portale kuri pateikiama pirmajame darbo su sistema puslapyje. Registracijos metu turi būti užpildomi pagrindiniai laukai kurie reikalingi vartotojo ir jo duomenų saugumui išlaikyti. Po sėkmingos registracijos, kiekvienas jau kartą buvęs prisijungęs vartotojas turi galimybę pamatyti savo praeitos sėkmingai užmegztos sesijos metu paliktą būseną. Pagal nutylėjimą vartotojui nematant ir visa su juo susijusi informacija ir inicijuotais pokalbiais užsaugoma sistemoje – šiuo atveju MySQL duomenų bazėje.

Autistiško vaiko įgūdžiai ir gebėjimai turi būti lygūs arba kiek žemesni nei tokių metų sveiko vaiko įgūdžiai naudojantis kompiuteriu ir programine įranga per internetą. Rekomenduojama pastovi auklėtojų arba tėvų priežiūra įveikiant darbo su šia sistema sunkumus.

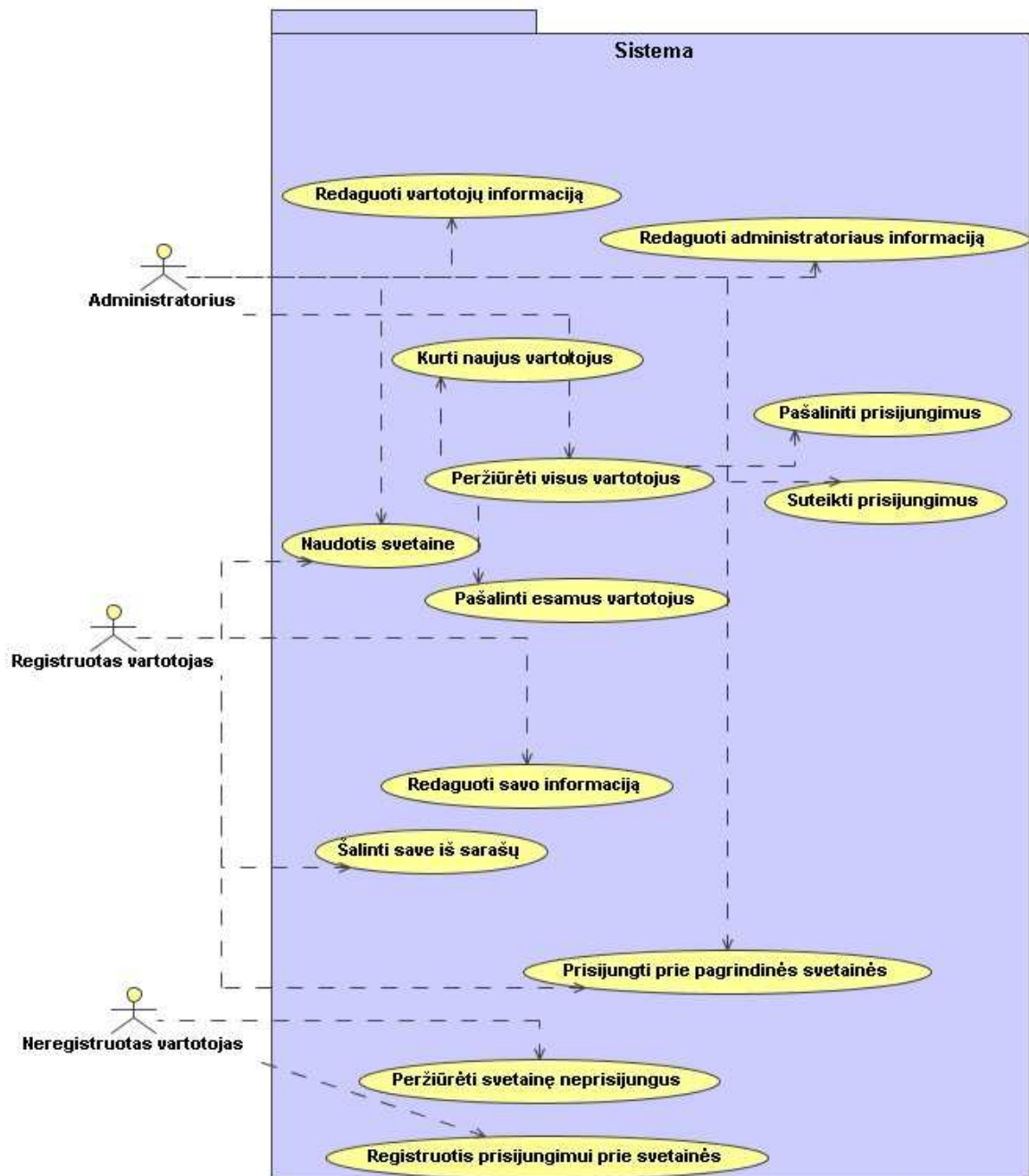
### 3.2.2. Vartotojų tipai ir teisės

Projekte pagal nutylėjimą numatyti trys vartotojų grupės. Administratorių, registruotų vartotojų ir portalo lankytojų. Daugiausia sistemos funkcionalumo gali išnaudoti administratorius kiek mažiau, registruotas vartotojas ir teisių beveik neturi neprisiregistravęs vartotojas. Vartotojų teisės pateikiamos 3 lentelėje.

3 lentelė. Vartotojų teisės

<b>Vartotojų tipas</b>	<b>Vartotojų grupės teisės</b>
<i>Administratorius</i>	<ul style="list-style-type: none"><li>- sukurti naujus vartotojus,</li><li>- pašalinti jau esamus vartotojus,</li><li>- suteikti prisijungimus,</li><li>- pašalinti prisijungimus,</li><li>- redaguoti vartotojų informaciją</li><li>- redaguoti savo informaciją</li><li>- peržiūrėti visus vartotojus</li><li>- prisijungti prie pagrindinės svetainės</li><li>- naudotis svetaine</li></ul>
<i>Registruotas vartotojas</i>	<ul style="list-style-type: none"><li>- redaguoti savo informaciją</li><li>- prisijungti prie pagrindinės svetainės</li><li>- naudotis svetaine</li><li>- šalinti save iš sąrašų</li></ul>
<i>Neregistruotas vartotojas</i>	<ul style="list-style-type: none"><li>- registruotis prisijungimui prie svetainės</li><li>- peržiūrėti svetainę neprisijungus</li></ul>

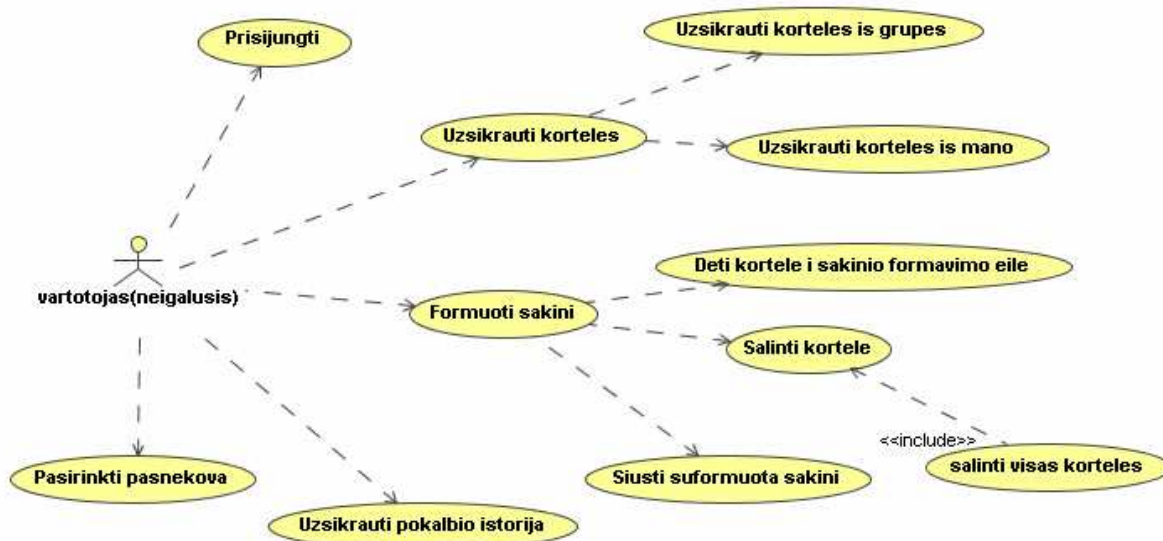
Vartotojų panaudojimo diagrama (angl. *use-case*) pateikta žemiau (5 pav.).



5 pav. Vartotojų teisių panaudojimo (use-case) diagrama

Su vartotojų teisėmis ir grupėmis glaudžiai susijusios statistikos bei registravimo žurnalų posistemiai. Kaupiant šių posistemų duomenis svarbu, ar vartotojas identifikuojamas ar identifikuojami jo sistemos parametrai, nes nuo to priklauso saugumo lygis svetainėje ir vartotojų asmeninė informacija.

Žemo lygio pagrindinio sistema besinaudojančio vartotojo panaudojimo atvejų diagrama pavaizduota paveiksle žemiau (6 pav.).



6 pav. Žemiausio lygio neįgalaus vartotojo veiksmai

Panaudojimų atvejų specifikacijos:

<b>Panaudojimo atvejis</b>	<i>prisijungti</i>
<b>Numeris</b>	1
<b>Aktorius</b>	Vartotojas(neįgalusis), Administratorius
<b>Sistema</b>	Neįgaliųjų abilitacijos svetainė
<b>Prieš sąlyga</b>	Sistemos vartotojas turi įvesti prisijungimo duomenis
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas įveda jungimosi duomenis 2. Vartotojas keičia slaptažodį 3. Vartotojas jau yra prisijungęs	1.1. Sistema patikrina įvestus duomenis, jeigu jie teisingi tuomet suteikia vartotojui teises darbui su sistema..  1.2. Sistema patikrina įvestus duomenis, jeigu jie teisingi, tuomet suteikia administratoriui teises darbui su sistema.  2. Sistema patikrina prisijungimo duomenų teisingumą ir leidžia vartotojui pasikeisti esamą prisijungimo slaptažodį į naują.
<b>Po sąlyga</b>	Suteikiamas priėjimas prie sistemos.
<b>Alternatyvos (nesėkmės atvejai)</b>	1.1.a. Blogas prisijungimo vardas. Sistema išveda klaidos pranešimą, grįžta į 1 žingsnį 1.1.b. Blogas prisijungimo slaptažodis. Sistema išveda klaidos pranešimą, grįžta į 1 žingsnį 2.1. Ne vienodai du kartus įvestas naujas slaptažodis. Sistema išveda klaidos pranešimą.
<b>Vykdymo variantai</b>	1. Vartotojas įveda prisijungimo duomenis 2. Vartotojas jau yra prisijungęs
<b>Veiklos taisyklės</b>	Vartotojas turi teisingai įvesti duomenis.
<b>Specialūs (nefunkciniai) reikalavimai</b>	Vartotojas prieš pradėdamas naudotis sistema privalo prisiregistruoti prie sistemos.
<b>Kitos sistemos, su kuriomis sąveikauja sistema vykdydama PA</b>	-

Ryšiai su kitais PA	-
Pastabos	-
Neišspręstos problemos	-

<b>Panaudojimo atvejis</b>	<i>Užkrauti korteles</i>
<b>Numeris</b>	2
<b>Aktorius</b>	Neįgalusis
<b>Sistema</b>	Neįgaliųjų abilitacijos svetainė
<b>Prieš sąlyga</b>	Vartotojas turi būti prisijungęs prie sistemos
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas atlieka kortelių užkrovimą pagal jų grupes 2. Nurodo neteisingą kortelių kodą	1. Sistema ekrane pateikia korteles naudojamam sistemoje 2. Sistema paskaičiuoja korteles ir padaro puslapiavimą ekrane jei to reikia
<b>Po sąlyga</b>	Kortelės užkraunamos
<b>Alternatyvos (nesėkmės atvejai)</b>	1. Nerasta duomenų apie tokios grupės korteles. (Neišvedama ekrane).
<b>Vykdyto variantai</b>	1. Vartotojas įveda prisijungimo duomenis. 2. Vartotojas atlieka kortelės grupės pasirinkimą.
<b>Veiklos taisyklės</b>	-
<b>Specialūs (nefunkciniai) reikalavimai</b>	Vartotojas turi būti identifikuotas kaip registruotas vartotojas.
<b>Kitos sistemos, su kuriomis sąveikauja sistema vykdydama PA</b>	-
<b>Ryšiai su kitais PA</b>	Užkrauti korteles iš grupės, užkrauti korteles iš mano
<b>Pastabos</b>	-
<b>Neišspręstos problemos</b>	-

<b>Panaudojimo atvejis</b>	<i>Formuoti sakinį</i>
<b>Numeris</b>	3
<b>Aktorius</b>	Neįgalusis
<b>Sistema</b>	Neįgaliųjų abilitacijos svetainė
<b>Prieš sąlyga</b>	Vartotojas turi būti prisijungęs prie sistemos
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Įvedami du simboliai reikalingi sistemai atpažinti korteles 2. Spaudžiama ant kortelės	1. Vartotojas susiranda reikiamą kortelę duomenis 2. Įveda kortelę į sakinio formavimo eilę
<b>Po sąlyga</b>	Į DB įvesta ir ekrane parodoma kortelės turinys
<b>Alternatyvos (nesėkmės atvejai)</b>	1. Įvedamos ne toks kortelės kodas arba paspaudžiama ant tuščios kortelės. 2. Sistemai nepavyko išsaugoti sulyginimo.
<b>Vykdyto variantai</b>	1. Vartotojas įveda prisijungimo duomenis. 2. Vartotojas įveda vieną ar kelias korteles 3. Užsaugomos kortelės(automatiškai)
<b>Veiklos taisyklės</b>	-

<b>Specialūs (nefunkciniai) reikalavimai</b>	Vartotojas turi būti identifikuotas kaip registruotas vartotojas. Kodus privaloma įvedinėti pagal nustatytą formato standartą.
<b>Kitos sistemos, su kuriomis sąveikauja sistema vykdydama PA</b>	-
<b>Ryšiai su kitais PA</b>	Dėti kortelę į sakinio formavimo eilę, šalinti kortelę, siūsti suformuotą sakinį.
<b>Pastabos</b>	-
<b>Neišspręstos problemos</b>	-

<b>Panaudojimo atvejis</b>	<i>Užkrauti pokalbio istoriją</i>
<b>Numeris</b>	4
<b>Aktorius</b>	Neįgalusis
<b>Sistema</b>	Neįgaliųjų abilitacijos svetainė
<b>Prieš sąlyga</b>	Vartotojas turi būti prisijungęs prie sistemos
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. vartotojas suranda ir paspaudžia istorijos dalyje 2. Uždaroma istorijos dalis	1. Sistema užkrauna istoriją su dabartinio vartotojo pašnekėsio ankstesne istorija 2. Tęsiamas darbas su sistema
<b>Po sąlyga</b>	Parodyta istorija
<b>Alternatyvos (nesėkmės atvejai)</b>	-
<b>Vykdymo variantai</b>	1. Vartotojas įveda prisijungimo duomenis. 2. Pažiūrima istorija
<b>Veiklos taisyklės</b>	-
<b>Specialūs (nefunkciniai) reikalavimai</b>	Vartotojas turi būti identifikuotas kaip registruotas vartotojas. Privalu žinoti kur rasti ar ieškoti pokalbių istorijos lango
<b>Kitos sistemos, su kuriomis sąveikauja sistema vykdydama PA</b>	-
<b>Ryšiai su kitais PA</b>	-
<b>Pastabos</b>	-
<b>Neišspręstos problemos</b>	-

<b>Panaudojimo atvejis</b>	<i>Pasirinkti pašnekovus</i>
<b>Numeris</b>	5
<b>Aktorius</b>	Neįgalusis
<b>Sistema</b>	Neįgaliųjų abilitacijos svetainė
<b>Prieš sąlyga</b>	Vartotojas turi būti prisijungęs prie sistemos
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. vartotojas suranda ir paspaudžia pašnekovą su kuriuo norės bendrauti 2. inicializuojamas pokalbis	1. Sistema parenka vartotoją ir jam nusiunčiamas pranešimas apie bendravimo pradžią 2. Sistema siunčia ir gauna pranešimus iš vartotojo
<b>Po sąlyga</b>	Vartotojas parenkamas
<b>Alternatyvos (nesėkmės atvejai)</b>	-



<b>Vykdyimo variantai</b>	1. Vartotojas įveda prisijungimo duomenis. 2. Parenkamas prisijungęs pašnekovas
<b>Veiklos taisyklės</b>	-
<b>Specialūs (nefunkciniai) reikalavimai</b>	-
<b>Kitos sistemos, su kuriomis sąveikauja sistema vykdydama PA</b>	-
<b>Ryšiai su kitais PA</b>	-
<b>Pastabos</b>	-
<b>Neišspręstos problemos</b>	-

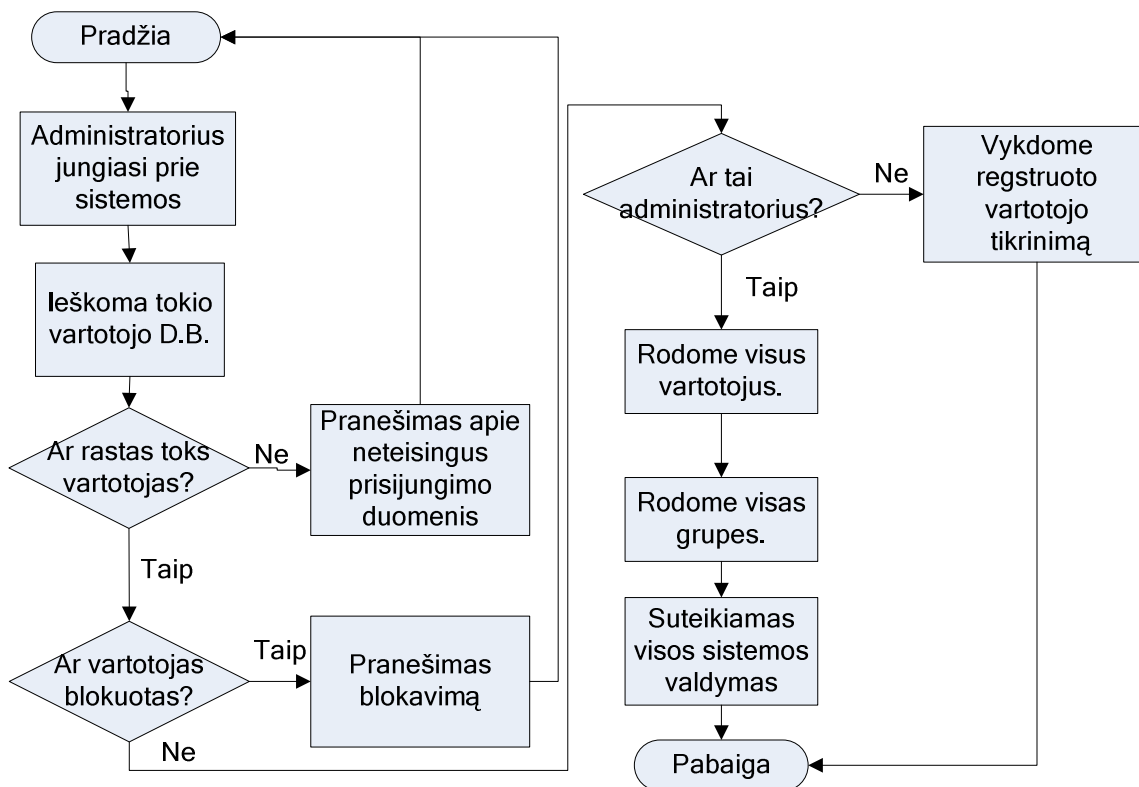
### 3.2.3. Vartotojų autorizavimas

Vartotojų prisijungimas vykdomas iš visų būsenų, kurios aprašytos prieš tai buvusiame skyriuje. Kiekvienai būsenai patikrinamas loginis teisingumas ir iš gautų rezultatų vertinama ar leisti, ar atmesti vartotojo prisijungimo prašymą. Vartotojų prijungimą galima aprašyti dviem algoritmais kurie priklauso nuo vartotojo tipo.

Vartotojų prisijungimo algoritmas, kai jungiasi pagrindinis administratorius:

- jungiasi prie sistemos;
- jeigu prisijungimo duomenys teisingi vartotojas pamato valdymo sistemos langą, kitu atveju vartotojui parodomas pranešimas ir siūloma jungtis vėl;
- jei tai pagrindinis administratorius prijungiamame prisijungus prie sistemos mato visus jam suteiktus valdyti tinklo kompiuterius, grupes, vartotojus ir taisykles.

Prisijungimo algoritmo kai jungiasi administratorius, ar registruotas vartotojas schema matoma žemiau (7 pav.).



7 pav. Vartotojų jungimosi prie sistemos veiklos diagrama

Prisijungimas galimas iš pagrindinio prisijungimo sistemoje lango suvedus tikslus savo duomenis ir paspaudus „Enter“ arba „prisijungti mygtuką“. Po kiekvieno nepavykusio prisijungimo vartotojas gauna šansą jungtis dar kartą. Saugumo sumetimais vartotojų slaptažodžiai saugomi MD5 kodavimo algoritmais.

### 3.3. Reikalavimai sistemai

#### 3.3.1. Funkciniai sistemos reikalavimai

Kuriamai sistemai keliami šie funkciniai reikalavimai:

- *Patogumas* – sistema nevalgintų ir būtų patogi, paprasta ir intuityvi naudoti tai vienas svarbiausių kriterijų į kurią dedama daug darbo ir pastangų. Galima sakyti sistemos funkcionalumas netoks svarbus kaip patogumas, nes trumpam reikia save pastatyti į neįgaliojo vietą ir pabandyti išsivaizduoti sistemos panaudojimo galimybes
- *Funkcionalumas* – būtų įgyvendintos visi funkciniai apibrėžtumai.
- *Vaizdinės informacijos gausa* – vaizdinė informacija pateikiama grupuotai ir todėl jos galima pateikti daug.

- *Panaudojamumas* – norint pasinaudoti sistema reikalingas personalinis kompiuteris ir interneto prieiga.
- *Suprantamumas* – projektuojant ir netgi kuriant turi būti įvertinta kad pagrindinis vartotojas bus vaikas ir jo suvokimo lygis bus kiek žemesnis už vidutinio vaiko suvokimo lygį.
- *Naudingumas* – sistemos nauda pasireiškia savo unikalumu ir informacinių technologijų specifiniu pritaikymu,
- *Stabilumas* – svarbu išvengti kritinių sisteminių programavimo klaidų kurių metu būtų sunaikinti vartotojų duomenys ar įvykdytas nesankcionuotas programos veikimas.
- *Praplečiamumas* – sistemos naudojimo sėkmės atveju būtina numatyti galimą sistemos praplečiamumą.

### 3.3.2. Nefunkciniai sistemos reikalavimai

Kuriamai sistemai keliami šie nefunkciniai reikalavimai:

- *Naujumas* – naujumas projekte įgyvendinamas per pritaikymo sferą ir per netradicinę projekto auditoriją. Būtent tokio analogo atliekančio specifinį funkcionalumą dar nėra.
- *Išbaigtumas* – sistemos galutiniam veikimo principui įgyvendinti šis reikalavimas nėra labai svarbus nes numatomas tokios sistemos funkcionalumo papildymas.
- *Greitumas* – kaip ir kiekvienos sistemos paremtos veikimu per internetą taip pat ir šios sistemos į tai atsižvelgti būtina, bet to užtikrinimas priklauso nuo labai daug faktorių. Iš sisteminės pusės įmanomas optimizavimas kai kurių sistemos dalių, kad kaip įmanoma daugiau mažinti tinklo srautą, taip reikalingą vaizdinei medžiagai - privaloma.
- *Naujų technologijų panaudojimas* – jis svarbus tuomet kai siekiama sukurti visiškai naują produktą, o šiuo atveju norima pagerinti naudojimosi patogumą ir išvengti situacijos kai neįgalusis negalės pasinaudoti visomis sistemos glimybėmis vien dėl to kad ne ten bus sutelktas funkcionalumas
- *Tarptautiniai suderinamumai* – Pradinė šios sistemos versija turi būti realizuota darbu viena kalba - Lietuvių. Numatoma galimybė papildyti sistemą, kad vėlesnės versijos galėtų veikti su kitomis kalbomis.

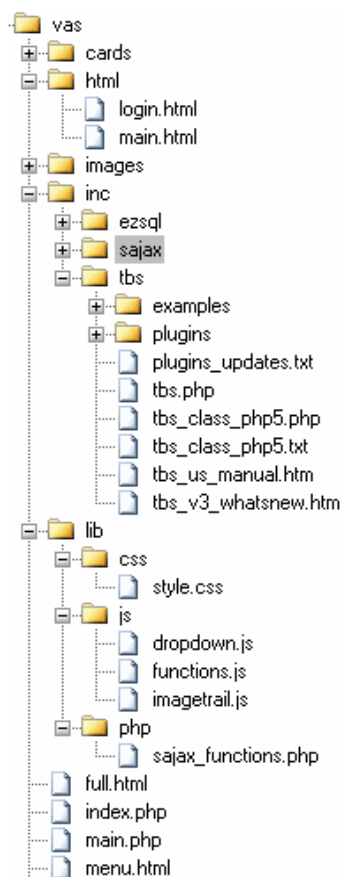
### 3.4. Diegimo aplinka

Įdiegimas vyksta perkeliant visus programinius failus į interneto serverį, todėl reikalingas kvalifikuotas asmuo atlikti šį darbą. Duomenys laikomi duomenų bazės MySQL serveryje. Taip pat būtina išmanyti serverio duomenų bazės struktūrą, ir turėti vienos ar visų lentelių administravimo teises. Projekto failuose, pagrindiniame jo kataloge yra suformuotas .sql failas kurio dėka lengvai galima suimportuoti lentelių, struktūras ryšius ir duomenis. Norint sėkmingai įdiegti sistemą prie jos būtina pritaikyti duomenų bazę – suvedant reikiamus duomenis į programos šakniniame kataloge esantį \*.cfg.php plėtinio failą.

Paleidžiamieji ir pagrindiniai sisteminiai failai:

- „index.php“ – vartotojų prisijungimo forma, vartotojų meniu ir registracijos langai. Jis paleidžiamas naudojant formatą „http://(svetaines ar sub-domeno pavadinimas kuriame yra programa)/[index.php]“. Šiuo atveju index.php yra nebūtinas nebent serverio konfigūracijoje pagal nutylėjimą nėra nurodyti indekso failai dažniausiai tai („index.html“, „index.htm“, „index.php“, „index.php4“ ir k.t.)
- „main.php“ – šiuo failu paleidžiamas ir vykdomas pagrindinis programos langas. Jo paleidimu nereikia specialiai rūpintis, kadangi saugumo sumetimais visas paleidimas inicijuojamas per „index.php“ failą pasirinkus teisingą vartotojo vardą ir slaptažodį.
- „database.cfg.php“ failas kurio pagalba duomenų bazė supranta kokį prisijungimo vardą ir slaptažodį naudoti bei prie kokios duomenų bazės jungtis. Failas specialiai sugalvotas saugumo sumetimais su dvigubu plėtiniu „cfg.php“. Taip neleidžiamas nepageidaujamas duomenų bazės perėmimas ir kontroliavimas, kadangi serveryje veikia php-deamonas kuris pamatęs tokio failo plėtinį iškarto jį inicijuoja. Rezultate gaunamas baltas, tuščias naršyklės langas.

Failų sistema turi būti sukurta (arba perrašyta) taip kaip pavaizduota paveiksle žemiau (8 pav.).



**8 pav. Sistemos failų struktūra**

Vartotojai prie sistemos jungsis naudodami internetinę naršyklę, tiesiog įvesdami reikiamą URL ir suveddami administratoriaus patvirtintus vartotojo vardą ir slaptažodį.

### ***3.5. Reikalavimai operacinei sistemai***

#### **3.5.1. Kliento operacinė sistema**

Sistemos paleidimui operacinė sistema įtakos neturi. Svarbiausia, jog operacinė sistema kurioje norima paleisti šią informacinę sistemą palaikytų standartinę(grafinę) interneto naršyklę (geriausia vieną iš šio metu populiariųjų interneto naršyklių Internet Explorer 6+, Firefox 1.5+, Opera 8+ ir t.t.) ir joje būtų sudiegtos visos naujausios apsaugos nuo sistemos piratavimo priemonės bei kritiniai operacinės sistemos atnaujinimai. Programai skirtos interneto naršyklės pasirinkimas nulėmė apklausų rezultatai surinkti iš interneto vartotojų balsų. Statistiniai paskutinių trijų metų duomenys pateikti žemiau (9 pav.)

<b>2006</b>	<b>IE7</b>	<b>IE6</b>	<b>IE5</b>	<b>Fx</b>	<b>Moz</b>	<b>N7/8</b>	<b>O7/8/9</b>
October	3.1%	54.5%	3.2%	28.8%	2.4%	0.3%	1.4%
September	2.5%	55.6%	4.0%	27.3%	2.3%	0.4%	1.6%
August	2.0%	56.2%	4.1%	27.1%	2.3%	0.3%	1.6%
July	1.9%	56.3%	4.2%	25.5%	2.3%	0.4%	1.4%
June	1.6%	58.2%	4.3%	24.9%	2.2%	0.3%	1.4%
May	1.1%	57.4%	4.5%	25.7%	2.3%	0.3%	1.5%
April	0.7%	58.0%	5.0%	25.2%	2.5%	0.4%	1.5%
March	0.6%	58.8%	5.3%	24.5%	2.4%	0.5%	1.5%
February	0.5%	59.5%	5.7%	25.1%	2.9%	0.4%	1.5%
January	0.2%	60.3%	5.5%	25.0%	3.1%	0.5%	1.6%
<b>2005</b>	<b>IE6</b>	<b>IE5</b>	<b>Ffox</b>	<b>Moz</b>	<b>N7</b>	<b>O8</b>	<b>O7</b>
December	61.5%	6.5%	24.0%	2.7%	0.4%	1.3%	0.2%
November	62.7%	6.2%	23.6%	2.8%	0.4%	1.3%	0.2%
October	67.5%	6.0%	19.6%	2.6%	0.4%	1.2%	0.2%
September	69.8%	5.7%	18.0%	2.5%	0.4%	1.0%	0.2%
August	68.4%	6.3%	18.9%	2.4%	0.4%	0.8%	0.3%
July	67.9%	5.9%	19.8%	2.6%	0.5%	0.8%	0.4%
June	65.0%	6.8%	20.7%	2.9%	0.6%	0.7%	0.5%
May	64.8%	6.8%	21.0%	3.1%	0.7%	0.7%	0.6%
April	63.5%	7.9%	20.9%	3.1%	0.9%	0.4%	1.0%
March	63.6%	8.9%	18.9%	3.3%	1.0%	0.3%	1.6%
February	63.9%	9.5%	17.9%	3.3%	1.0%		1.7%
January	64.8%	9.7%	16.6%	3.4%	1.1%		1.9%
<b>2004</b>	<b>IE6</b>	<b>IE5</b>	<b>Moz</b>	<b>N3</b>	<b>N7</b>	<b>N4</b>	<b>O7</b>
December	65.5%	9.9%	17.0%	0.2%	1.2%	0.2%	1.8%
November	66.0%	10.2%	16.5%	0.2%	1.2%	0.3%	1.6%
October	67.3%	10.8%	14.7%	0.3%	1.3%	0.3%	1.6%
September	67.8%	11.2%	13.7%	0.3%	1.4%	0.3%	1.7%
August	67.0%	13.0%	12.7%	0.4%	1.4%	0.4%	1.6%
July	67.2%	13.2%	12.6%	0.4%	1.4%	0.4%	1.6%
June	67.6%	13.2%	12.2%	0.5%	1.4%	0.4%	1.6%
May	68.1%	13.8%	9.5%	0.6%	1.4%	0.4%	1.6%
April	68.2%	14.0%	8.5%	0.8%	1.4%	0.6%	1.4%
March	68.2%	14.6%	7.9%	0.8%	1.4%	0.6%	1.4%
February	68.3%	15.2%	7.3%	0.6%	1.5%	0.4%	1.5%
January	68.9%	15.8%	5.5%	0.4%	1.5%	0.5%	1.5%

9 pav. Interneto naršylių naudojimas. Šaltinis [26]

Sistema veikia tik operacinėje sistemoje kuri palaiko grafinę sąsają (angl. GUI). Tai reiškia nors operacinė sistema veikia išskirtinai tik konsoliniu režimu (tokios OS kaip Unix, Linux), o taip pat palaiko ir konsolinę interneto naršyklę (pvz. Lynx) tai sistemos veikimas tokioje aplinkoje neįmanomas.

Kaip jau minėta sistema veikia per interneto prieigą, todėl operacinė sistema turi sugebėti leisti prisijungti (palaikyti užpildymo laukų dialogų sistemą), ir parsisiųsti reikalingas daugialypės terpės bylas į kompiuterį.

Nors ir neapibrėžta tiksli operacinė sistema tačiau nepatyrusiam kompiuterių vartotojui rekomenduojama Windows XP operacinė sistema ir aukštesnės šios platformos versijos. Ji turėtų turėti naujausią egzistuojančią interneto naršyklę kuri palaiko bent šiuos internetinius standartus: HTML 4, DOM 1 ir JavaScript 1.2 versijas.

### 3.5.2. Serverio operacinė sistema

Serverio operacinė sistema nesvarbi. Ji turi atitikti šiuolaikinių internetinio serverio standartus ir palaikyti reikiamus protokolus. Operacinės sistemos serveris turi leisti php prisijungimus ir interpretavimus, palaikyti MySQL duomenų bazę. Maksimalus prisijungimų skaičius priklauso nuo to kiek serveryje bus laikinosios atminties(RAM). Vidutiniškai normali operacinė sistema su naujausiomis DB bibliotekomis palaiko nuo 500 iki 1000 prisijungimų vienu metu priklausomai nuo to ką veikia vartotojai. Jei serveryje rodoma duomenų bazės klaida dėl „per daug prisijungusių vartotojų“, tai galima pataisyti programiškai nukreipiant vartotojų srautą į kitą nutolusį serverį.

### 3.6. Reikalavimai įrangai

Rekomenduojamo kompiuterio duomenys:

- procesorius (CPU) : nuo 500 MHz ir daugiau;
- operatyvinė atmintinė (RAM): nuo 128 MB ir daugiau;
- kietasis diskas (HDD): nuo 20GB ir daugiau;
- grafinė atmintinė(VRAM): nuo 1MB ir daugiau;
- optinis įrenginys(CD-ROM): nebūtinai;
- lanksčiųjų diskelių įrenginys: nebūtinai;
- USB įrenginys: reikalingas
- tinklo plokštė

Tai ne pats optimaliausias variantas, tačiau šios sistemos pakanka normaliam IS darbui. Šiuo metu kuriant programinius produktus stengiamasi daugiau naudoti greito programavimo metodus ir logikos elementus negu visiškai apkrauti sistemą perkrautomis ir nereikalingomis funkcijomis. Nustatytas operacinės sistemos apribojimas kuris turi būti išsamiai patikrintas ir suderintas su čia sukurta sistema naudoja mažų sisteminių rodiklių aparatūrinę įrangą Didesni įrangos duomenys gali smarkiai viršyti tuos kurie nurodyti rekomenduojamoje sistemoje, nuo to sukurtos programinės sistemos darbas ir jos rezultatai nepriklauso.

### 3.7. Kritiniai sistemos faktoriai

Sistema funkcionuoja teisingai:

- Sistema yra įdiegta,

- duomenų bazė kaupia atsiunčiamus duomenis,
- sistemos grafinis režimas neišdarkytas,
- sistemos darbas interneto tinkle užtikrintas,
- sistema reaguoja į klavišų ir pelės kombinacijas.

Interfeisas:

- Suprantamas galiniam vartotojui (angl. *end-user*),
- nevargina,
- nenuteikia prieš norą mokintis.

Testavimas:

- Atliktas kruopštus sistemos testavimas,
- nerasta kritinių klaidų.

### 3.8. Išvados

- Sudaryti sisteminiai įrangos ir funkciniai bei nefunkciniai reikalavimai. Pastebėta, kad tokios sistemos reikalavimai pakankamai aukšti ir užsibrėžtą tikslą pasiekti labai sunku, bet įmanoma.
- Daug laiko praleista prie operacinių sistemų ir jų savybių parinkimų, atitinkančių sisteminius duomenis, bei prisidedančius prie programos išpildymo.
- Aprašytos egzistuojančių vartotojų teisės ir nustatyti keliami reikalavimai jiems. Atsižvelgta, jog pagrindinis vartotojas į kurį orientuota sistema turi būti ir yra neįgalusis vaikas.



## 4. Sistemos aprašas

### 4.1. *Techninė užduotis*

#### 4.1.1. Varotojo sąsaja

Sukurti ir realizuoti (suprogramuoti) informacinę sistemą veikiančią interneto portale. Padaryti ją prieinamą ir naudojamą fizinę negalią turintiems neįgaliesiems, suprantantiems ir dalinai suprantantiems lietuvių kalbą.

Jos valdymas turi būti suprantamas ir intuityvus. Valdymas turi būti realizuotas bent jau standartine kompiuterio pele ir klaviatūra. Kiekvienas veiksmas gali būti atliekamas pasirinktinai vien klaviatūra arba vien pele, bet galimas ir abiejų įrenginių valdymo derinys.

Svetainės realizavimas turi būti informacinio – žaidybinio pobūdžio. Svetainės turinys neturi sukelti priešiško jausmų ir nepageidaujamų emocijų. Spalvos sistemoje turi būti parinktos kuo šviesesnės ir kuo neutralesnės – geriausia iš pastelinės gamos derinio. Ekranų spalvinis fonas gali būti raminantis, bet kartu neatitraukiantis vartotojo sutelkto dėmesio nuo jo pagrindinio tikslo šioje svetainėje.

Svetainės vartotojui kiekvienas jo veiksmas neturi asocijuotis su sunkiai įvykdoma užduotimi t.y. valdymas turi būti kuo paprastesnis ir kuo lengviau suprantamas šio specifinio tipo vartotojui.

Nereikalingas nereikšmingų objektų sureikšminimas spalviniais deriniais juos galima išskirti tik šių komponentų naudojimo metu.

#### 4.1.2. Operacinės sistemos

Sukurta veikimui internete ji turi būti nepriklausoma nuo operacinių sistemų (Windows ar Linux) platformų. Jos naudojimo metu turi pakakti normalaus OS funkcionalumo t.y. nieko papildomo nereikia įdieginti į nuosavą PK. Sėkmingam informacinės sistemos naudojimui nereikalingos jokios administratoriaus teisės ir papildomos kompiuterinės žinios, tik pagrindiniai kompiuterinio raštingumo žinios.

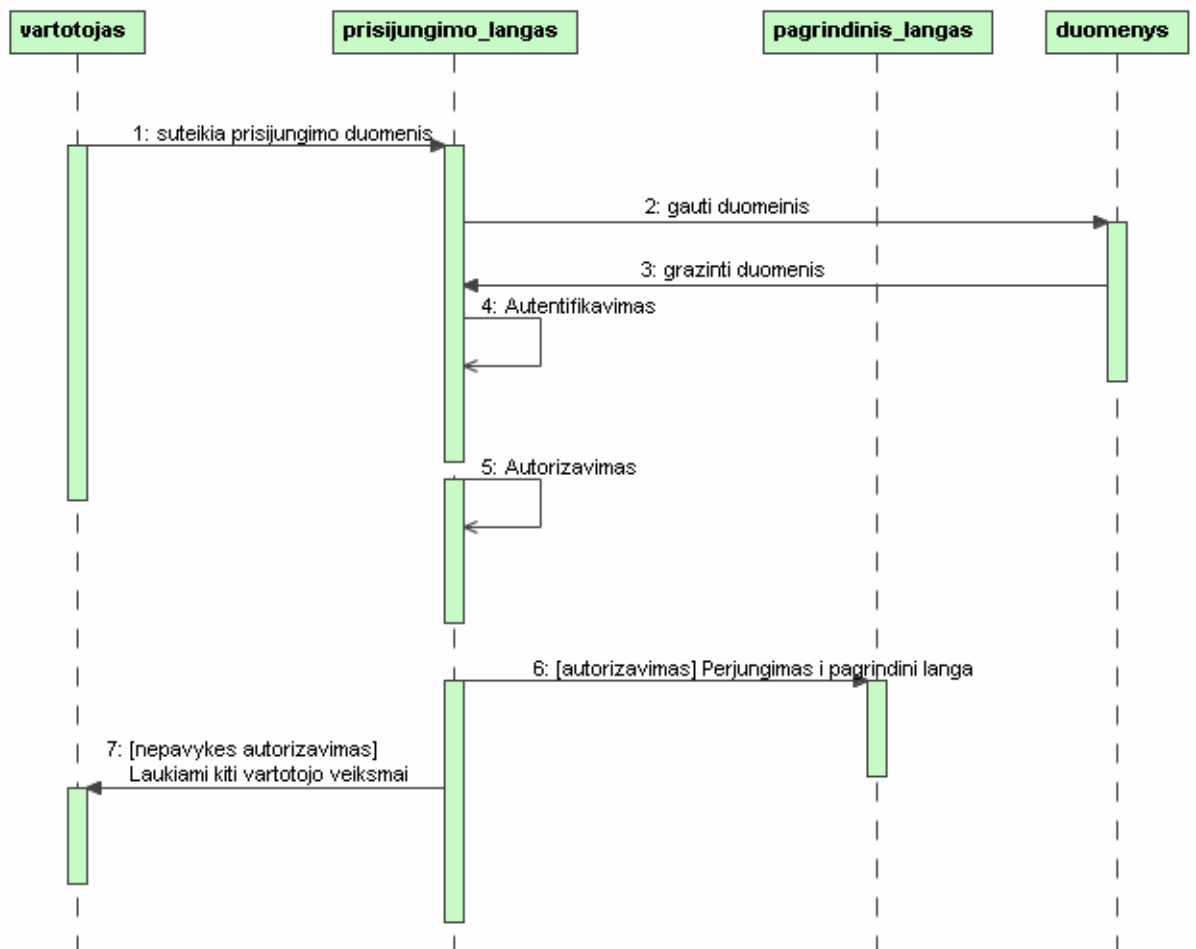
Naudojant operacinę sistemą kaip serverį jis turi būti pagrįstas nemokamomis licenzijomis paremtais serveriais. bet priklausoma nuo operacinių sistemų naršyklių.

### 4.1.3. Funkcionalumas

Igyvendinti reikalavimų specifikacijoje numatytą funkcionalumą, ir svetainės veikimo principus, pritaikant vaizdinių kortelių katalogus ir saugoti juos pagal kiekvieno vartotojo surenkamą pokabių statistiką. Sekti kurie vartotojai yra prisijungę prie sistemos ir kas gali šnekėtis ir kas ne. Nenukrypti nuo funkcinų reikalavimų.

### 4.1.4. Autentifikavimas

Kiekvienas sistemos vartotojas atpažįstamas pagal jam suteiktą vartotojo vardą ir į sistemą prijungiamas tik jam registracijos metu suteiktu vartotojo vardu ir slaptažodžiu. Neregistruoti vartotojai naudotis sistema negali tol kol neužsiregistruoja patys arba kol jų neužregistruoja sistemą administruojantis žmogus. Prisijungimas prie sistemos pavaizduotas žemiau (10 pav.).



10 pav. Vartotojo autentifikavimo ir autorizavimo sekų diagrama

#### 4.1.5. Realizavimas

Pasirinktos realizavimo priemonės turi būti nepriklausomos nuo operacinės sistemos. Pageidaujamos PHP, HTML ir JavaScript(AJAX). Vartotojo sesijos duomenims saugoti pasirinkta duomenų bazė MySQL.

Kadangi tiesiogiai iš AJAX sukuriama darbo aplinkos saugoti duomenų į MySQL duomenų bazėje techniškai neįmanoma, todėl tam pasirinkta PHP programavimo kalbos skriptai tarpininkauti įgyvendinant šį funkcionalumą. Realizavimo pasirinkimas neturi įtakoti aprašyto sistemos funkcionalumo ar vartotojo sąsajos nukrypimą.

### 4.2. Sistemos projektas

#### 4.2.1. Architektūra

Tarnybinės stoties, kurioje bus patalpinti pagrindiniai sistemos moduliai, techniniai duomenys priklausys nuo sistemos vartotojų skaičiaus, duomenų apimties bei vartojamos papildomos operatyvinės bei techninės įrangos.

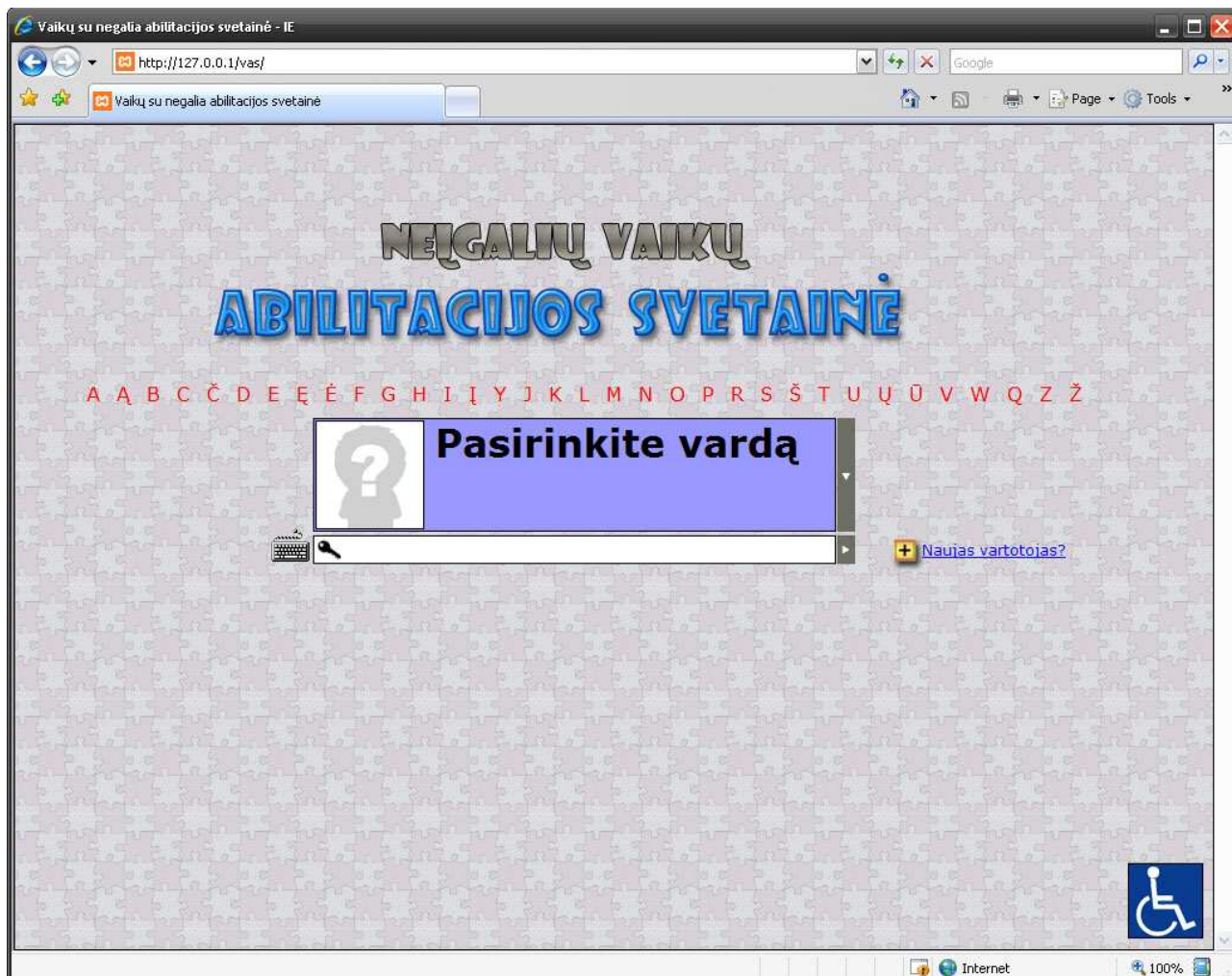
Tarnybinėje stotyje turi būti įdiegta tokia programinė įranga:

- Windows operacinė sistema (Windows 2000, Windows XP, Windows NT), Linux, Solaris, Sun
- Web serveris (Apache, IIS, kitas HTTP serveris)
- MySQL serveris
- PHP programavimo kalbos interpretatorius (PHP 4 arba vėlesnės versijos)

#### 4.2.2. Vizualus sistemos modelis

Programa sukurta naudojantis patarimais iš tokia veikla užsiimančių svetainių. Dauguma tokio tipo svetainių naudoja jau senai naudojamus kūrimo metodus kurie neįgaliajam žmogui turi palengvinti naudojimąsi internetiniu puslapiu. Tačiau labai retai kuri pasinaudoja naujomis technologijomis tokiomis kaip AJAX. Pagrindinis tokių sistemų, naudojančių naujas technologijas, privalumas tai, jog svetainės veikimas paremtas tik daliniu duomenų atnaujinimu. Taip taupomi interneto resursai, pagreitėja interneto puslapis, pagerėja patogumas. Šios technologijos labai plačiai pritaikytos ir šiame darbe.

Čia kuriama IS susideda iš dviejų pagrindinių langų reikalingų užsibrėžtam funkcionalumui įgyvendinti. Vizualus sistemos modelis ir pagrindinis prisijungimo langas pateikiamas žemiau (11 pav.).



11 pav. Pagrindinis prisijungimo langas

Visa ši internetinė sistema išsiskiria savo panaudojimo atvejų gausa t.y. ja galima sėkmingai naudotis tiek kompiuterine pele arba tiek klaviatūra (be pelės). Tai tikrai palengvina darbą neįgaliesiems turintiems įvairaus tipo motorinių sutrikimų. Visa svetainė tarsi didelis, klaviatūros paspaudimų laukiantis, langas. Kiekviena klaviatūros raidinė reikšmė (nuo A – Ž) turi paskirtį.

Šiame lange naudojama technologija leidžianti vartotojui sumažinti klaviatūros paspaudimų skaičių leidžiant išsirinkti savo prisijungimo vardą iš iškrentančio meniu. Paspaudus ant savo vardo pirmosios raidės arba ją įvedus klaviatūra automatiškai atidaromas iškrentantis meniu su šios raidės vardais surūšiuotais pagal abėcėlę. Tokio veiksmo įgyvendinimas, kai paspausta (arba įvesta) raidė pavaizduotas žemiau (12 pav.).

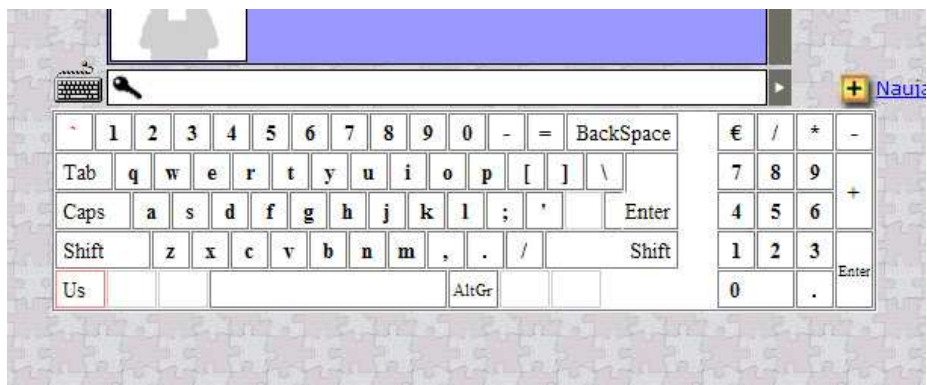


12 pav. Sistemos vartotojo pasirinkimas (paspausta „D“ raidė)

Paspaudus ant norimo vartotojo jis išsirenkamas į meniu viršutinę dalį po to meniu užsidaro. Sumaišius raides, veiksmus galima kartoti neribotą kartų skaičių nuo to prisijungimo kokybė nepriklauso. Padidėjus tokiu pačiu vardu iškrentančių vartotojų sąrašui automatiškai atsiranda vertikali slinkties juosta leidžianti judėti aukštyn ir žemyn ir taip susirasti savo vardą.

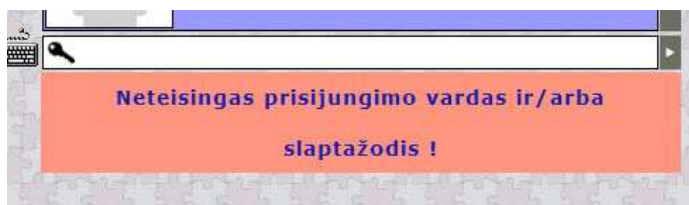
Jei vartotojas nerandamas arba jis nebuvo čia įvestas galima susikurti naują vartotoją spaudžiant „Naujas vartotojas“ nuorodą. Pasirodžiusioje registracijos formoje, užpildžius reikiamus duomenis vartotojas jau įvedamas į duomenų bazę ir gali prisijungti naudodamasis aukščiau pateikiama instrukcija.

Suradus savo vartotoją kaip ir registruojantis kitur, reikalinga suvesti savo prisijungimo slaptažodį. Jį taip pat galima suvesti pasirinktinai arba klaviatūra standartiškai arba pele. Slaptažodžio suvedimas pele yra kiek nestandartinis nes reikalinga pasinaudoti virtualia klaviatūra, kuri iškviečiama spustelėjus ant klaviatūros ikonėlės šalia arba spaudžiant „shift“ klavišą klaviatūroje. Klaviatūroje yra visi lotyniškos raidės, simboliai, bei skaitmenys kurie leidžia suvesti ir sugalvoti pakankamai sunkų slaptažodį naudojant visų šių trijų grupių aibes. Ekranas su klaviatūros realizacija pavaizduotas žemiau (13 pav.).



13 pav. Virtualios klaviatūros pritaikymas

Suvedus neteisingus prisijungimo duomenis t.y. suklydus pasirenkant vartotojo vardą arba suvedus neteisingai slaptažodį parodomas pranešimas apie tai. Pranešimas pavaizduotas žemiau (14 pav.).



14 pav. Pranešimas apie neteisingą prisijungimą

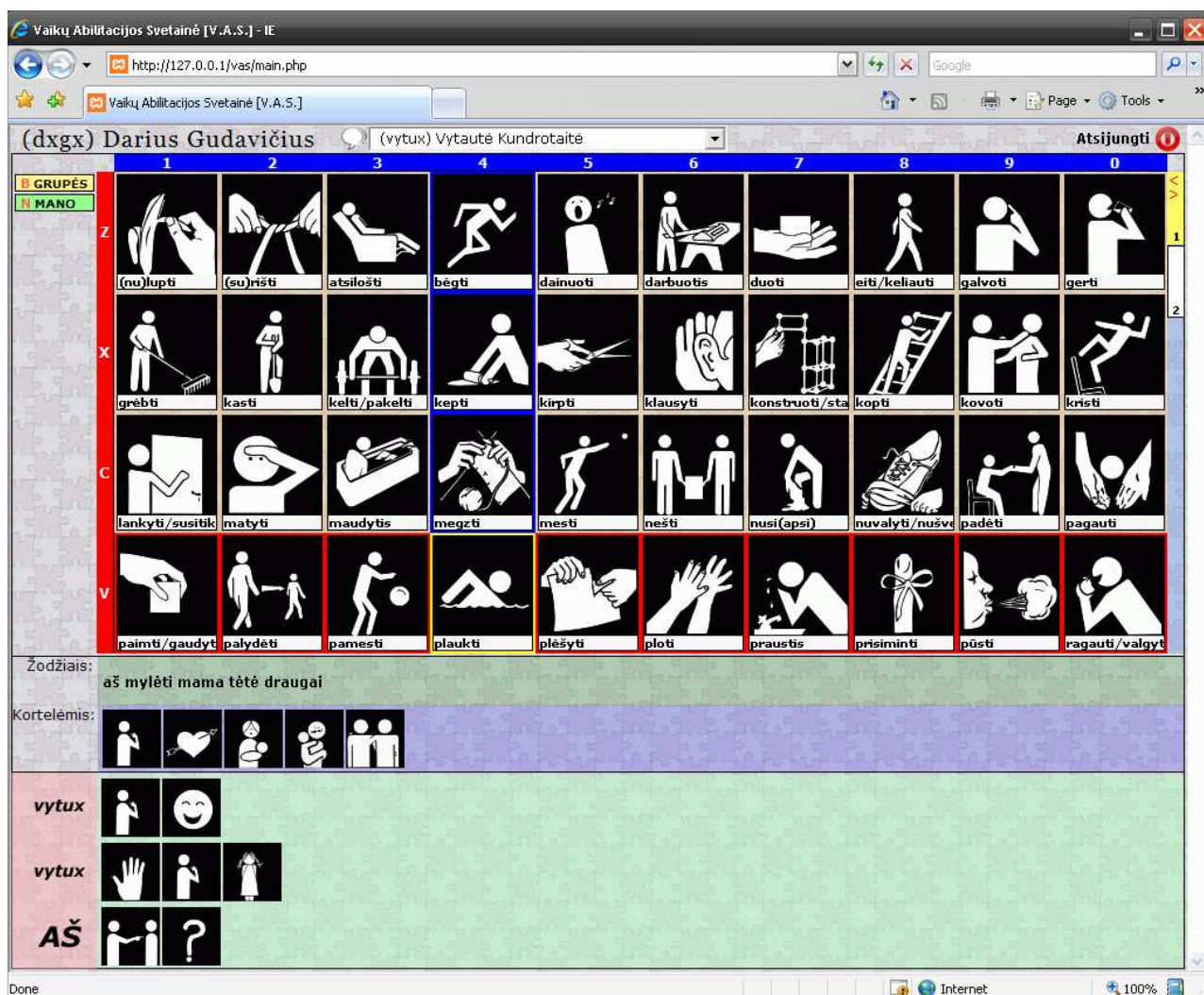
Jei duomenys visai nesuvesti(nepasirinktas vartotojo vardas arba neįvestas slaptažodis) pranešimo tekstas atrodo taip kaip pavaizduota paveiksle žemiau (15 pav.).



15 pav. Pranešimas apie neteisingus arba neužpildytus duomenis

Sistemoje vartotojai yra blokuojami, jei pastebėta nepageidautina ar kitokios formos neteisėta veikla. Tuomet jungiantis šiam vartotojui prie sistemos parodomas pranešimas „Šis vartotojas blokuotas prašome kreiptis į svetainės administratorių“ ir vartotojas negali toliau jungtis prie sistemos.

Sėkmingai prisijungus prie sistemos vartotojas perkeliamas į pagrindinį kortelių langą kuriame jis praleidžia daugiausia laiko. Šis langas yra ribotų matmenų (pritaikyta rezoliucijai 1024x768) todėl sutalpinti daug kortelių nėra paprasta. Vienu metu čia pateikiama maksimaliai 40 aktyvių kortelių kurių dėka įmanomas bendravimas. Pagrindinis sistemos langas pateiktas žemiau(16 pav.).



16 pav. Prisijungus matomas pagrindinis ekranas

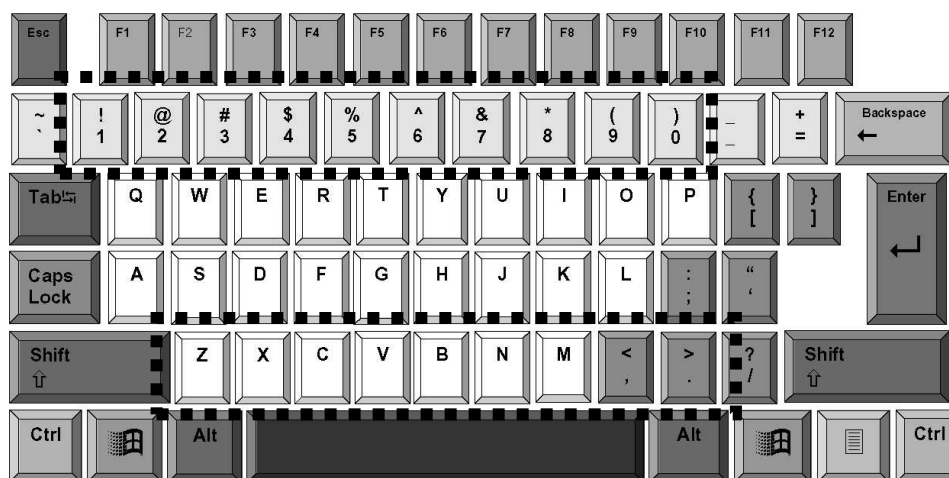
Vos prisijungęs vartotojas mato savo paskutinio pokalbio istoriją. Ten pat ją gali peržiūrėti platesniame lange užvedęs pelę ant raudonos vartotojų vardų vietos arba spaudžiant atitinkamą klaviatūros klavišą. Lange galiojančios klaviatūros reikšmės pateikiamos žemiau (4 lentelė).

4 lentelė. Aktyvūs klaviatūros klavišai lange

Klavišas	Paskirtis
Space	Išvaloma žodžių(kortelių) rinkimo eilė
BackSpace	Išvaloma žodžių(kortelių) eilės paskutinė kortelė
Enter	Siunčiama surinkta žodžių(kortelių) eilė
Shift	Atidaromas vartotojų galinčių šnekėti sąrašas
<	Peršokama į sekantį („aukštn“) kortelių puslapį(jei jų yra)
>	Peršokama į sekantį („žemyn“) kortelių puslapį(jei jų yra)
M	Istorijos lango iškvietimas(padidinimas)
Z	Parenkama pirma kortelių eilutė
X	Parenkama antra kortelių eilutė
C	Parenkama trečia kortelių eilutė

V	Parenkama ketvirta kortelių eilutė
B	„Grupių“ meniu iškvietimas
N	„Mano“ kortelių iškvietimas
1,2,3,4,5,6,7,8,9,0	Atitinkamų stulpelių parinkimas kortelių lentelėje
Esc	Išėjimas iš sistemos, analogiškai „atsijungti“ paspaudimas

Verta pažymėti, jog raidės Z,X,C,V,B,N,M,<, > buvo pasirinktos ne atsitiktinai. Jos klaviatūroje išdėstytos viena po kitos todėl tai padaro klaviatūrą patogią naudojimui. Kaip išdėstyti klavišai klaviatūroje pateikiama žemiau (17 pav.).



17 pav. Klaviatūroje išdėstyti valdymo klavišai

Be klaviatūros paspaudimų visas operacijas aprašytas aukščiau galima sėkmingai atlikti pasinaudojant pele. Taigi programos valdymas pasidaro lengvesnis jei nereikia naudoti abiejų rankų.

Svarbus aspektas yra tai kaip galima susidėti korteles pokalbiui ir jas nusiųsti pašnekovui. Pirmiausia reikia pasirinkti sau pašnekovą. Pašnekovai laukiantys pokalbio yra specialiame iškrentančiame meniu viršuje kuris kviečiamas paspaudus pele arba klaviatūra paspaudus „Shift“ klavišą. Pasirinkimo meniu matomas žemiau(18 pav.).



18 pav. Pasirinkimo meniu, puslapiavimas(dešinėje), atsijungimas(dešinėje, viršuje)

Čia parodomas visų prisijungusių prie sistemos surūšiuotas sąrašas. Sąraše nesimato jau kalbančių žmonių. Pasirinkus vieną žmogų iš sąrašo jam sistema automatiškai nusiunčia pranešimą



apie pokalbio pradžią užklaudama ar vartotojas norės šnekėtis kartu ar atsisakys šio pokalbio. Jei vartotojas spaudžia „Taip“ pokalbis inicijuojamas jei „Ne“ inicijavimas pasibaigia. Sistemoje numatytas faktorius jei abu vartotojai siunčia pranešimus vienas kitam tuomet pokalbio inicijavimas automatinis.

Pokalbio žodžiui sudaryti naudojama kortelių lentelė. Ji pavaizduota žemiau (19 pav.).

	1	2	3	4	5	6	7	8	9	0
Z	 (nu)lupti	 (cu)ršti	 atsilošti	 bėgti	 dainuoti	 darbuotis	 duoti	 eiti/keliauti	 galvoti	 gerti
X	 grėbti	 kasti	 kelti/pakelti	 kepti	 kirpti	 klausti	 konstruoti/sta	 kopti	 kovoti	 kristi
C	 lankyti/susitik	 matyti	 maudytis	 megzti	 mesti	 nešti	 nusi(apsi)	 nuvalyti/nušvę	 padėti	 pagauti
V	 paimti/gaudyt	 palydėti	 pamesti	 plaukti	 plėšyti	 ploti	 praustis	 prisiminti	 pūsti	 ragauti/valgyt

19 pav. Kortelių lentelė

Kortelių įdėjimo principas paprastas. Jei norima įdėti kortelę yra skaičiaus ir raidės susikirtime arba raidės ir skaičiaus susikirtime tai tereikia tą skaičių ir atitinkamai tą raidę paspausti. Kortelė esanti susikirtimo taške „įkrenta“ į žodžio formavimo eilę. Spaudžiant skaičių arba raidę atitinkamai kortelių stulpelis arba eilė pasižymi skirtingomis spalvomis, o jų susikirtimas išsiryškina dar kita pastebima spalva.

Kortelės netelpančios lange išdėstomos puslapiuotai. Puslapiavimas pavaizduotas aukščiau (18 pav.). Valdymo klavišai „<“, „>“ įgalina perjungti puslapius ir užkrauti kitas parinktos aktyvios grupės korteles.

Visa tai leidžia ir pelės veiksmai. Užvedus pelę ant kortelės ir paspaudus ant norimos ji perkeliama į žodžio formavimo eilę. Palaikius pelę kiek ilgiau ant kortelių, jos automatiškai išdidinamos.

Formuojant žodžius dažnai pritrūksta kortelių iš kitų grupių todėl jas taip pat galima perjunginėti pelės arba klavišo „B“ paspaudimu. Pasirodžiusiame meniu pateikiamos visos registruotos grupės su raidžių simboliais reiškiančiais, jog paspaudus atitinkamą raidę bus įkeliamos pasirinktos grupės kortelės. Meniu pavyzdys pateikiamas žemiau(20 pav.).



20 pav. „Grupės“ paspaudimo metu iškrenta meniu

Surinkus reikiamą kortelių skaičių kad sudaryti žodį jis išsiunčiamas „Enter“ klavišo paspaudimu aktyviam pokalbio draugui.

#### 4.2.3. Funkcijos ir funkcionalumas

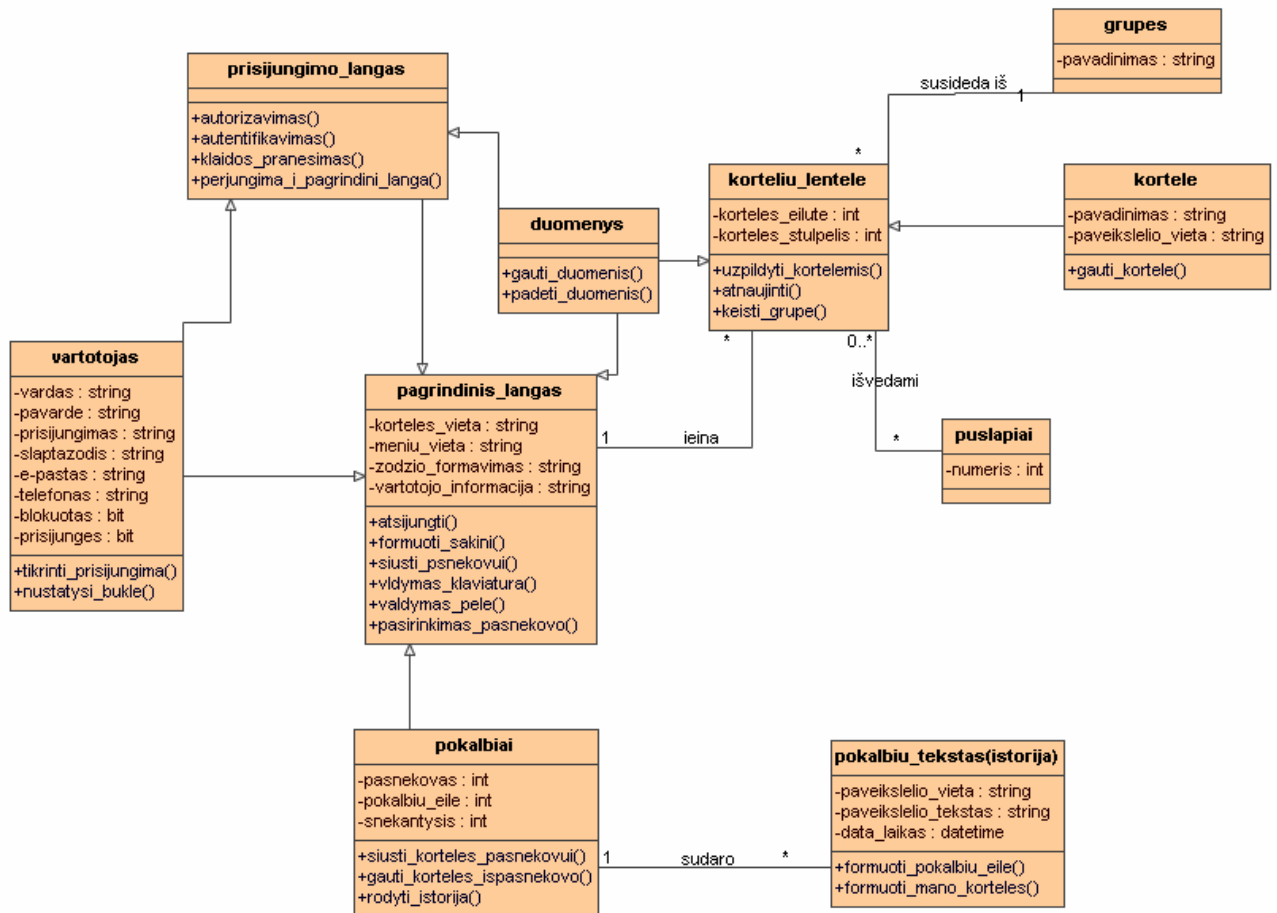
Pagrindinės sistemos funkcijos:

- sukurta neįgaliesiems;
- įgaliną bendravimą kortelėmis;
- leidžia bendrauti iš bet kurios pasaulio vietos;
- sistemos valgymas įmanomas naudojantis vien pele;
- sistemos valgymas įmanomas vien klaviatūra;
- naudoja mažai internetinių išteklių;
- saugo vartotojų duomenis (pokalbius);
- apsaugota nuo atsitiktinių vartotojo veiksmų (du kartus iš eilės negalima atlikti to paties veiksmo);
- pritaikyta vaikams;
- galimas kortelių vaizdo padidinimas;
- paryškkinama eilutė ir stulpelis;
- apdorojama daug kortelių;
- didesniai kortelių skaičiui įvedamas puslapiavimas;
- istorijos kaupimas;

- statistikos kaupimas ateičiai („Mano“ mygtuko funkcionalumas);
- virtualios klaviatūros panaudojimas;
- galima praplėsti duomenų bazę naujomis kortelėmis;
- administruojami vartotojai;
- vartotoju tarpusavio pokalbiai;

#### 4.2.4. Dalykinės srities klasių diagrama

Dalykinės srities klasių diagrama detaliai atvaizduoja kompiuterizuojamoje sistemoje naudojamas esybes bei jų tarpusavio ryšius. Jos paveikslas pateiktas žemiau (21 pav.)



21 pav. Klasių diagrama

*Vartotojas* – tai klasė skirta nusakyti vartotojo atributams ir tos klasės funkcijoms.

*Prisijungimo\_langas* - klasė skirta nusakyti prisijungimo lango ypatumams ir veiksmas su kitomis išorinėmis klasėmis ją papildo (angl. generalization) vartotojų ir duomenų klasės, o ji papildo pagrindinis\_langas klasę.

*Pagrindinis\_langas* – klasė nusakanti pagrindinio sistemos lango atributus ir funkcijas. Ją papildo vartotojas, pokalbiai, prisijungimo langas, duomenys, klasės, o ji siejasi ryšiu „vienas us daug“ korteliu\_lentele klase.

*Korteliu\_lentele* – tai pagrindiniame lange esanti kortelių matrica su metodais ir atributais kuriais naudojasi sistemoje. Ji susideda iš grupių, kortelių, ir puslapių klasių.

*Grupes* – tai kortelių grupės kurios įeina į korteliu\_lentelė klase.

*Korteles* – tai pačios kortelės nusakomos paveikslėliu ir pavadinimu. Ši klasė papildo korteliu\_lentele klase savo atributais ir funkcijomis.

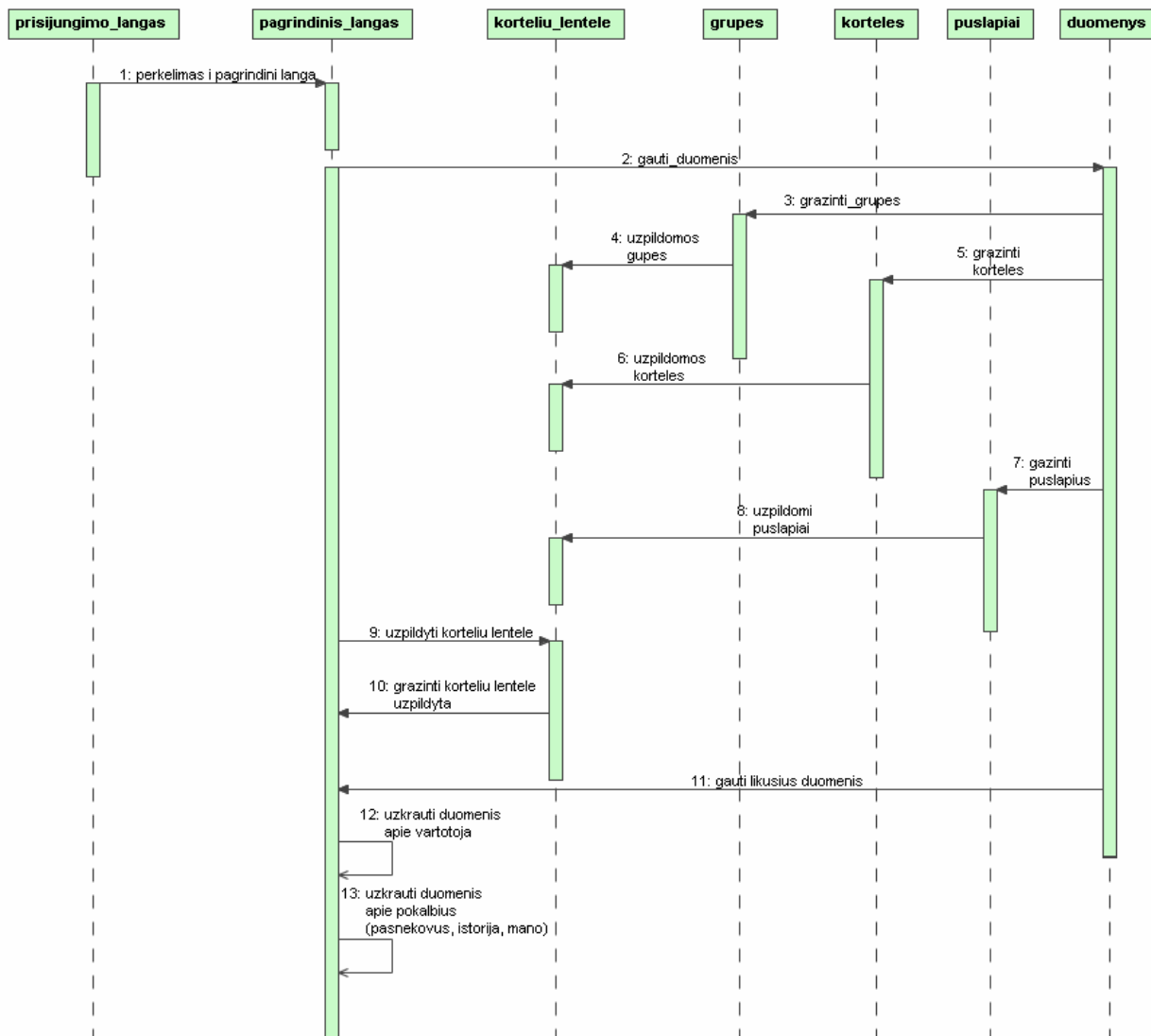
*Puslapiai* – puslapiavimui realizuota klasė. Ji siejasi ryšiu „daug arba nulis“ su korteliu\_lentele klase.

*Pokalbiai* – pokalbiams su kitu pašnekovu realizuoti skirta klasė ji nusako kaip ir kam rodomos atitinkamos kortelės. Ji susidaro iš pokalbiu\_tekstas klasės

*Pokalbiu\_tekstas* – nusakomi visi pasakyti tekstai, laikas, vieta, paveikslėlis ir paveikslėlio tekstas.

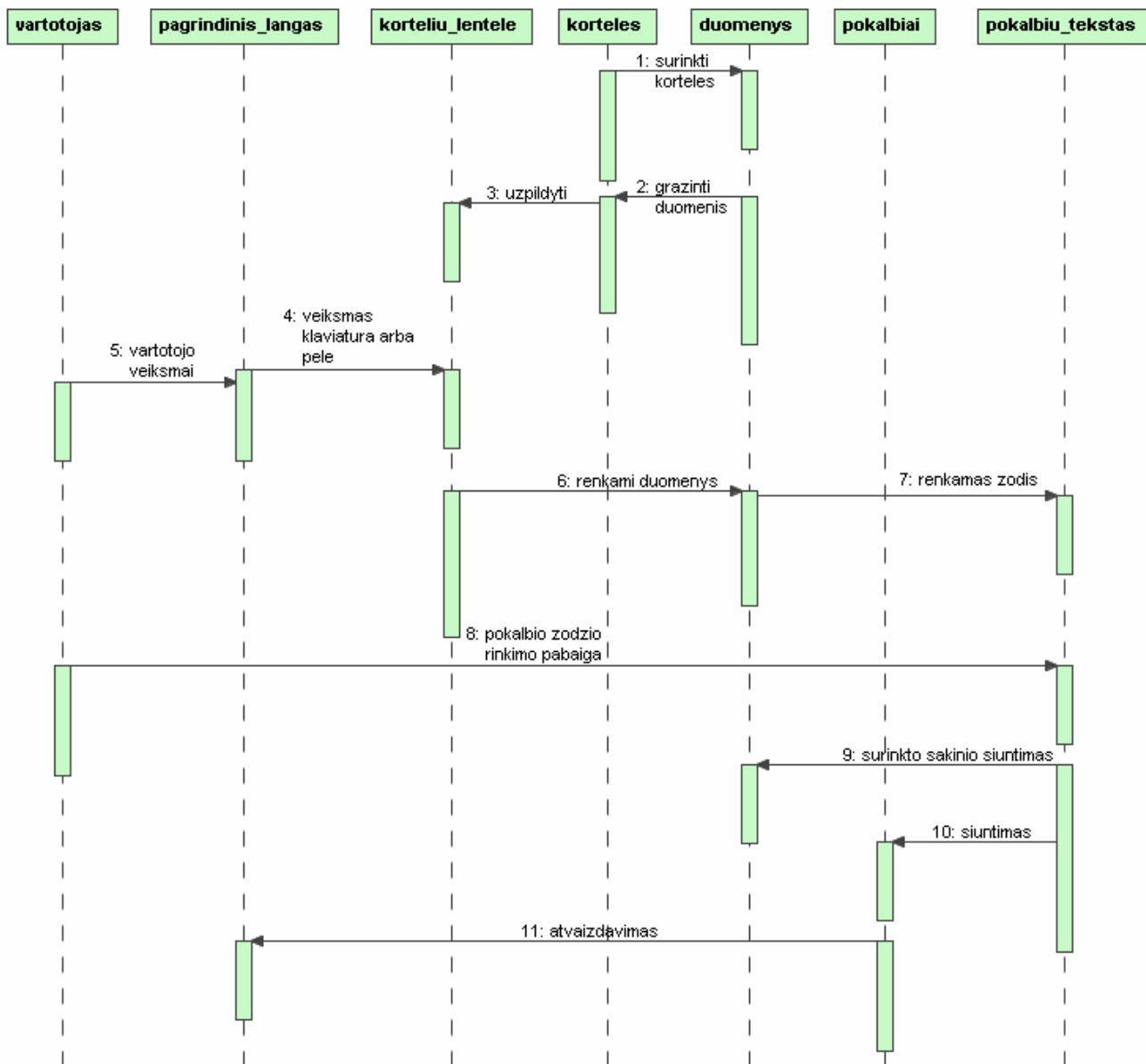
#### 4.2.5. Veikimo principai ir sekų diagramos

Pagrindinio lango ir visų jo dalių užkrovimo schema vaizduojama žemiau (22 pav.).



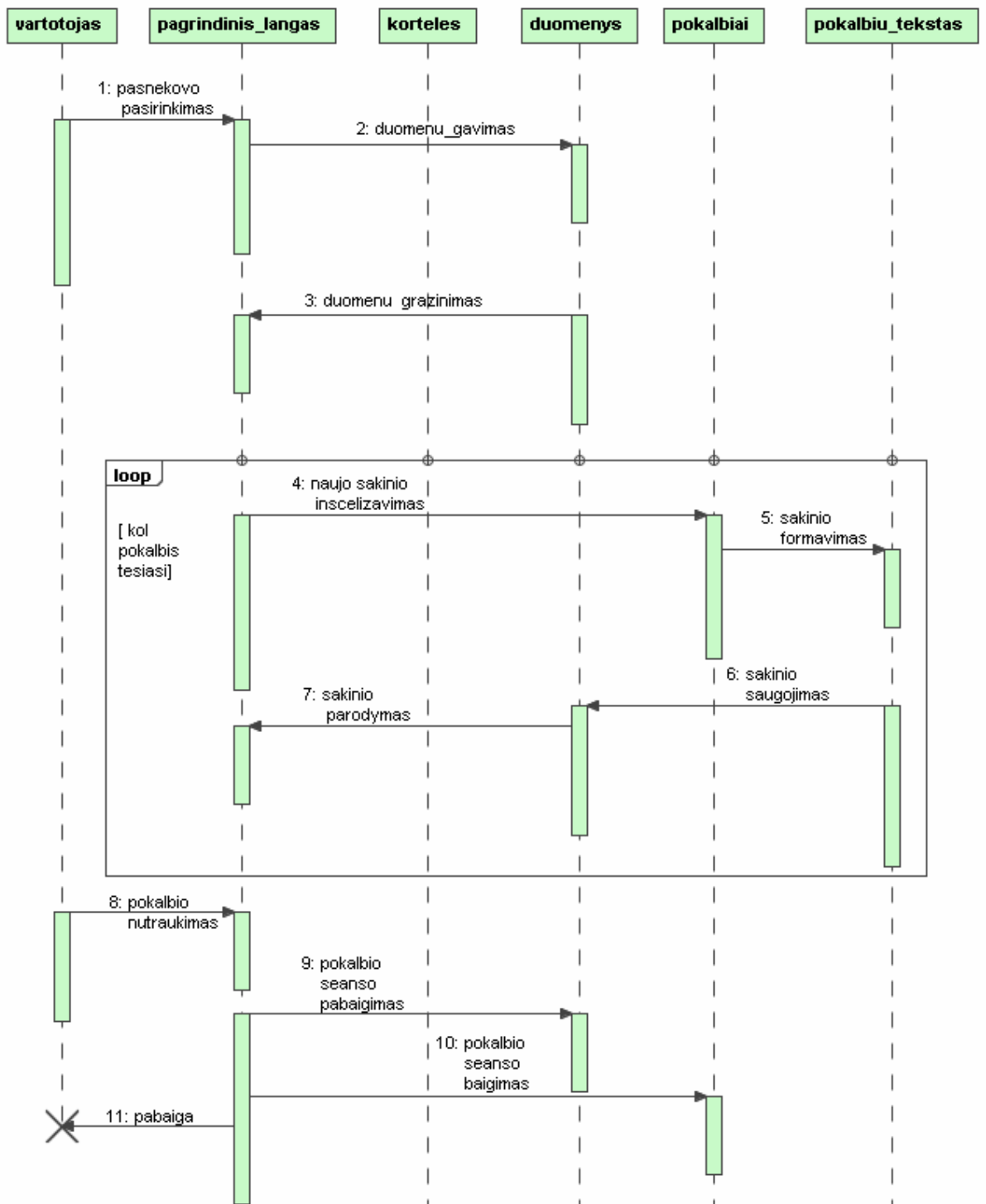
22 pav. Prisijungimo prie pagrindinio lango ir jo kortelių užkrovimo sekų diagrama

Kortelių įdėjimo į vartotojo sakinio formavimo eilę sekų diagrama pavaizduota žemiau (23 pav.).



23 pav. Kortelės įdėjimo (išėmimo) iš formavimo eilės sekų diagrama

Norint susiradus pašnekovą užmesti pokalbį su juo, sistemoje vykdoma ši sekų diagrama kuri pavaizduota žemiau (24 pav.)



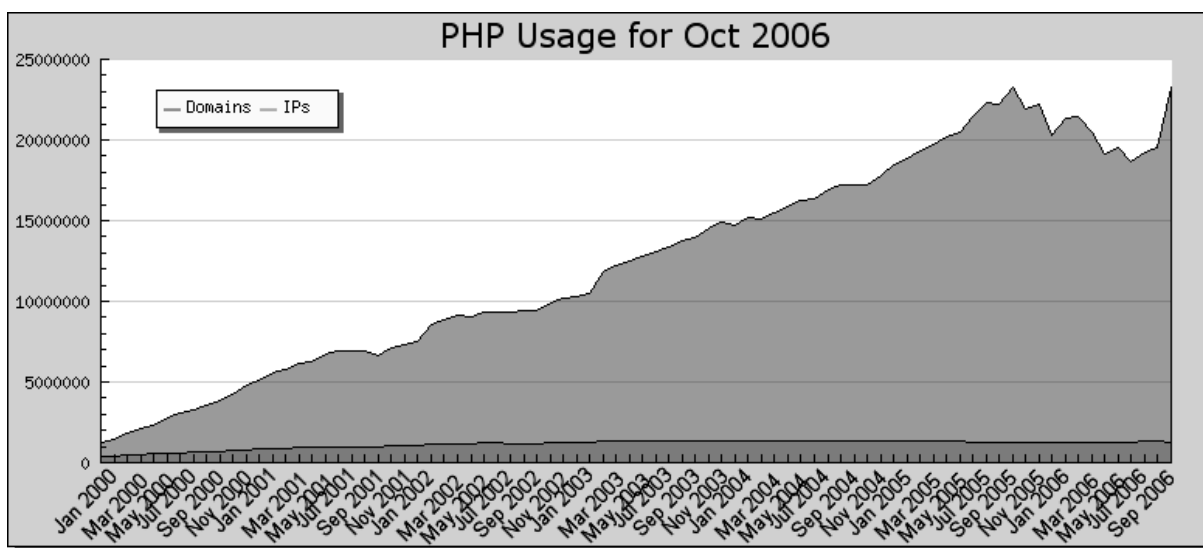
24 pav. Pokalbio inscenizavimo ir užbaigimo sekų diagrama

#### 4.2.6. Programavimo kalbos

Šis projektas buvo įgyvendintas panaudojant kelias programavimo, skriptų(angl. scripting language) kalbas daugiausia skirtas internetiniams projektams t.y. vartotojas-serveris (angl. client-server) sistemoms.

Visos programavimo kalbos yra platinamos nemokamai ar/ir laisvai įgyjamos internetu. Visi programoje sukurti skriptai su mintimi, jog bus naudojami vien tik nekomerciniams reikalams. Programavimo kalbos ir skriptų kalbos panaudotos projekte:

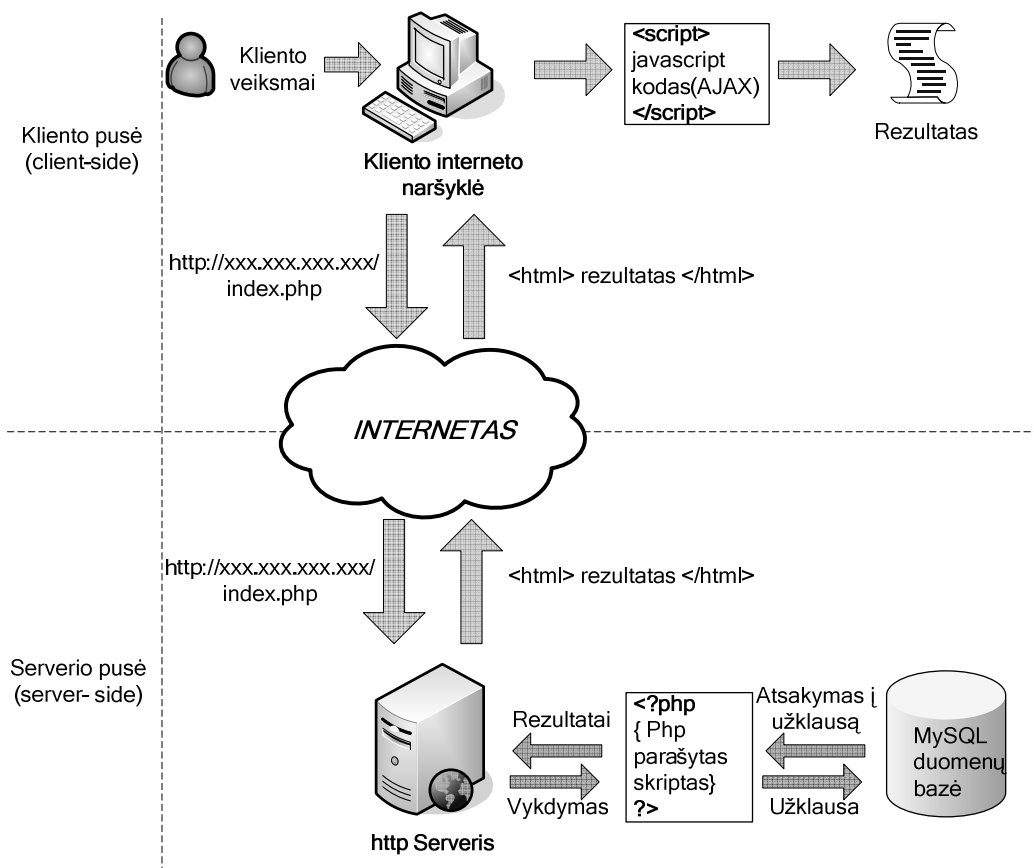
- PHP (angl. hypertext preprocessor) tai nemokama, atviro kodo, dinaminė interpretavimo-programavimo kalba kuri šiuo metu išpopuliarėjo ir aplenkė CGI programavimo kalbą savo lankstumu ir plačiu pritaikomumu su sisteminėmis komandomis. Šios programavimo kalbos naudotojų skaičiaus augimas pateikiamas žemiau (25 pav.)



25 pav. PHP naudojimo serveriuose statistika 2000 – 2006. Šaltinis [28]

Ji nesudėtinga bet lanksti – veikia daugumoje operacinių sistemų palaiko nemažai reliacinių duomenų bazių. Ši programavimo kalba tai maišytos struktūrinių C, Perl, Java kalbų sintaksės kūrinys, kuriuo dabar kuriami ne tik internetiniai produktai (kartu su HTML programavimo kalba ir MySQL duomenų bazėmis) bet ir savaeigės programos (angl. executable). Programa pasižymi naudinga savybe kuri nenaudoja jūsų kompiuterių resursų nes vykdoma serverio pusėje (angl. server side). Žemiau pateikta PHP, JavaScript, MySQL, HTML principinė veikimo ir duomenų apsikeitimo bendra schema pavaizduota 26 pav..





26 pav. Principinė PHP, JavaScript, MySQL, HTML veikimo schema

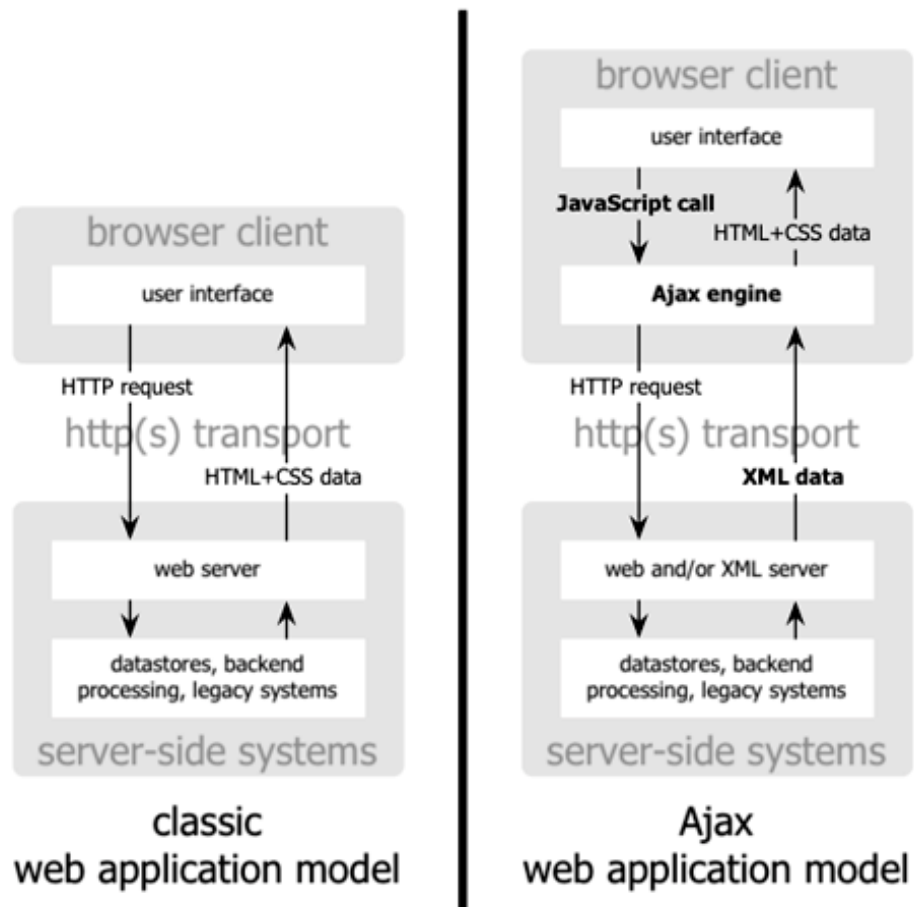
Schemos (26 pav.) veikimas prasideda kliento pusėje kai jis norėdamas savo interneto naršyklės lange pamatyti , jo pageidaujamą, tam tikrą internetinį puslapį. Tokiu būdu klientas savo naršyklės pagalba siunčia užklausa serveriui. Pasiųsta užklausa keliauja internetu nešdama kliento informaciją tol kol pasiekia reikiamą serverį ir čia jau ji pereina į serverio pusę (angl. server-side). Pasiekusi serverį užklausa analizuojama ar tikrai ji buvo čia siųsta jei taip ji pradeda interpretuoti ir jos kodas persiunčiamas PHP servisui kuris veikia jau šiame serveryje. Savo ruožtu PHP vykdo užklausoje pateiktas komandas vieną po kitos. Gali būti, kad programiniame kode yra ir susijungimo su duomenų baze realizavimas tuomet PHP jungiasi prie duomenų bazės, ten atlieka aprašytus veiksmus ir su sugražintais rezultatus toliau vykdo kodą. Kodo pabaigoje visa apdorota ir sugeneruota informacija perrašoma į HTML kodą su JavaScript ar kitomis komponentėmis kurį vėliau turės interpretuoti kliento pusėje esanti interneto naršyklė. Atlikus visus veiksmus vykdomas užklauso rezultatų grąžinimas iš serverio pusės į kliento pusę. Siunčiamas HTML failas grįžta tuo pačiu keliu kuriuo ir siuntė klientas. Čia jau kliento interneto naršyklė interpretuoja visą gražintą HTML kodą ir jame esančias JavaScript komponentes kurias apdoroja procesorius. Po paskutinės gauto kodo eilutės įvykdymo viskas parodoma klientui.

- XHTML(HTML), CSS. Tai internetinių puslapių kūrimo(žymėjimo) kalbos, kurios čia naudojamos kartu su PHP realizuoti puslapio dinamiškumą ir sukurti, grafinę vartotojo sąsają, valdymo aplinką, duomenų pateikimą, interneto naršyklėje.

HTML programavimo kalba įgalina puslapio pateikimą formuoti norimu pavidalu t.y. turinį pateikti kuo naudingiau išdėstant juos internetinės naršyklės lange. HTML savo standartu leidžia daug daugiau vienoms naršyklėms tačiau kitų teises nepaisydama. Todėl buvo sukurta XHTML kalba kuri turi beveik visas HTML savybes ir yra palaikoma daugeliu internetinių naršyklių dėl savo sukurto standarto universalumo. Ji daug daugiau orientuota dirbti su CSS stilių failais kurie įgalina puslapyje keisti spalvas, tekstus, fonus ir t.t.

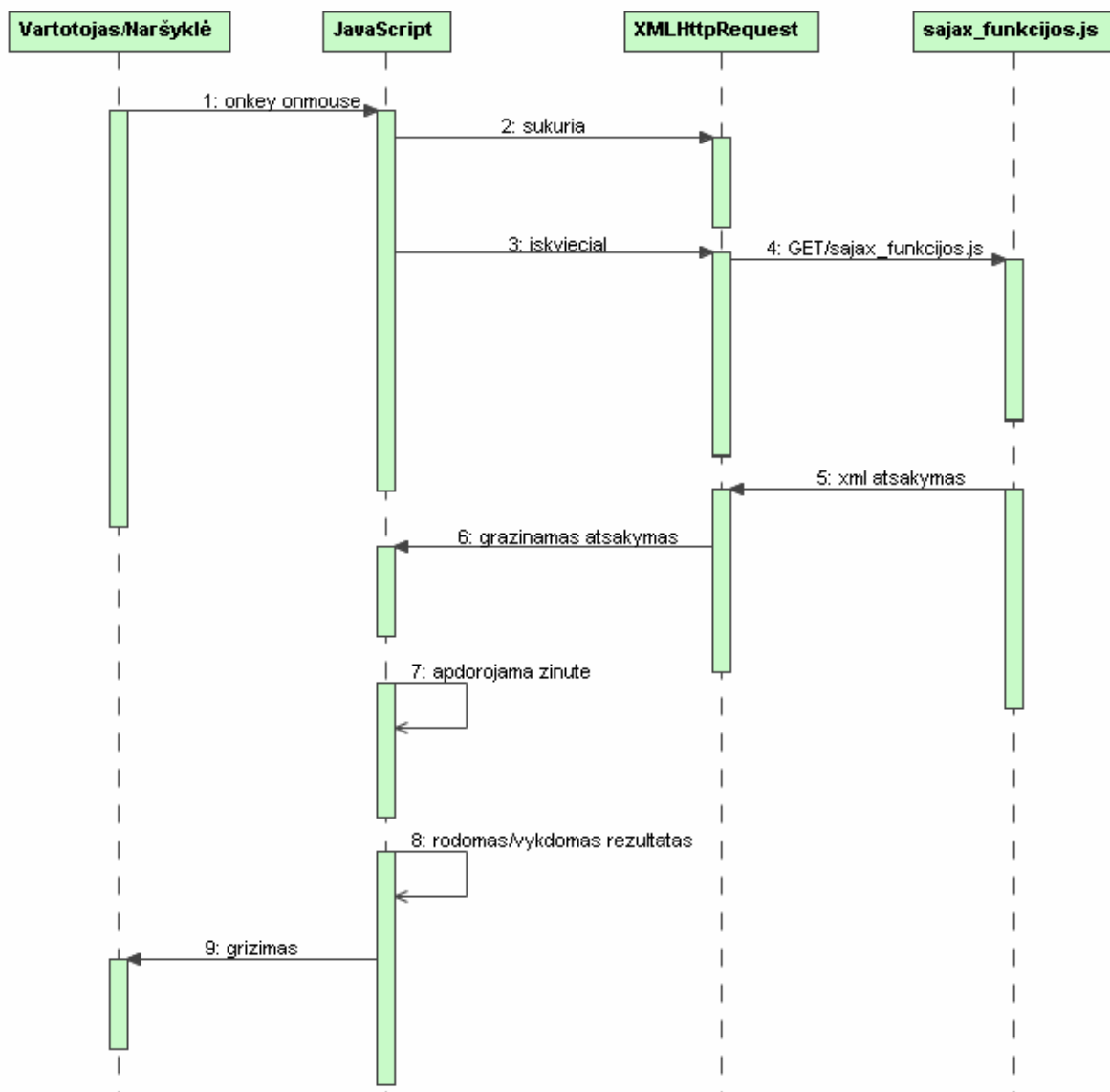
- JavaScript user-side (vykdoma kliento naršyklėje) programavimo kalba. Jos sintaksė labai panaši į PHP tačiau funkcionalumas iš esmės skiriasi. Ji orientuota į vartotojo naršyklės valdymą, apipavidalinimą, duomenų pateikimą, pranešimų išvedimą, nestandartinių įvedimo formų generavimą. Javascript dažniausiai naudojama internetinių puslapių interaktyvumo realizacijai, bet taip pat naudojama ir kaip galimybė skriptais manipuluoti tam tikromis programomis
- SQL pagrindinė programavimo kalba naudojama šiuolaikinėse reliacinėse ir ne tik, duomenų bazėse. Pagrindinis šios programavimo kalbos privalumas lengvai suprantama sintaksė ir lengvas užklausų formavimas sudėtingiems veiksams atlikti. SQL galimas dalinai paprastas eilučių pridėjimas, redagavimas, šalinimas, lentelių tarpusavio ryšių sukūrimas. Ji pasirinkta šiam projektui dėl lengvo susiejimo su PHP. Sistemos kūrimui naudojama konkretus duomenų bazės dialektas MySQL. Naujausia šios duomenų bazės versija(5.0+) palaiko vartotojo sukurtas funkcijas ir duomenų bazės kursorius(angl. cursors)
- AJAX (angl. Asynchronous JavaScript and XML) - Asinchroninė JavaScript skriptinio programavimo kalba ir XML žymėjimo kalba. Tai greičiau internetinio programavimo technologija nei programavimo kalba. Pagrindinis jos tikslas suteikti programuotojams laisvę kurti interaktyvius internetinius produktus ir web aplikacijas kurios leistų mažus duomenų apsikeitimus tarp serverio ir kliento kompiuterių vartotojui to net neįsijaučiant. Šios technologijos dėka nebereikia perkrauti interneto puslapių iš naujo. Taip pagreitinamas internetinio puslapio greitis, sudaromos sąlygos interaktyviai dirbti. Tokiu būdu puslapis tampa patogus ir greitas naudoti.

AJAX technologija susideda iš XHTML(arba HTML ir CSS), DOM(kuris pasiekiamas per JavaScript) bei pagrindinio XMLHttpRequest objekto naudojamo asinchronizuotam duomenų apsikeitimui įgyvendinti. Pagrindiniai skirtumai tarp klasikinio puslapio ir AJAX puslapio pateikti žemiau (27 pav.).



27 pav. Klasikinės ir Ajax internetinės aplikacijos modelis [30]

Ajax sekos diagrama sistemoje pavaizduota žemiau (28 pav.)



28 pav. Ajax sekos diagrama (naudojant javascript ir php)

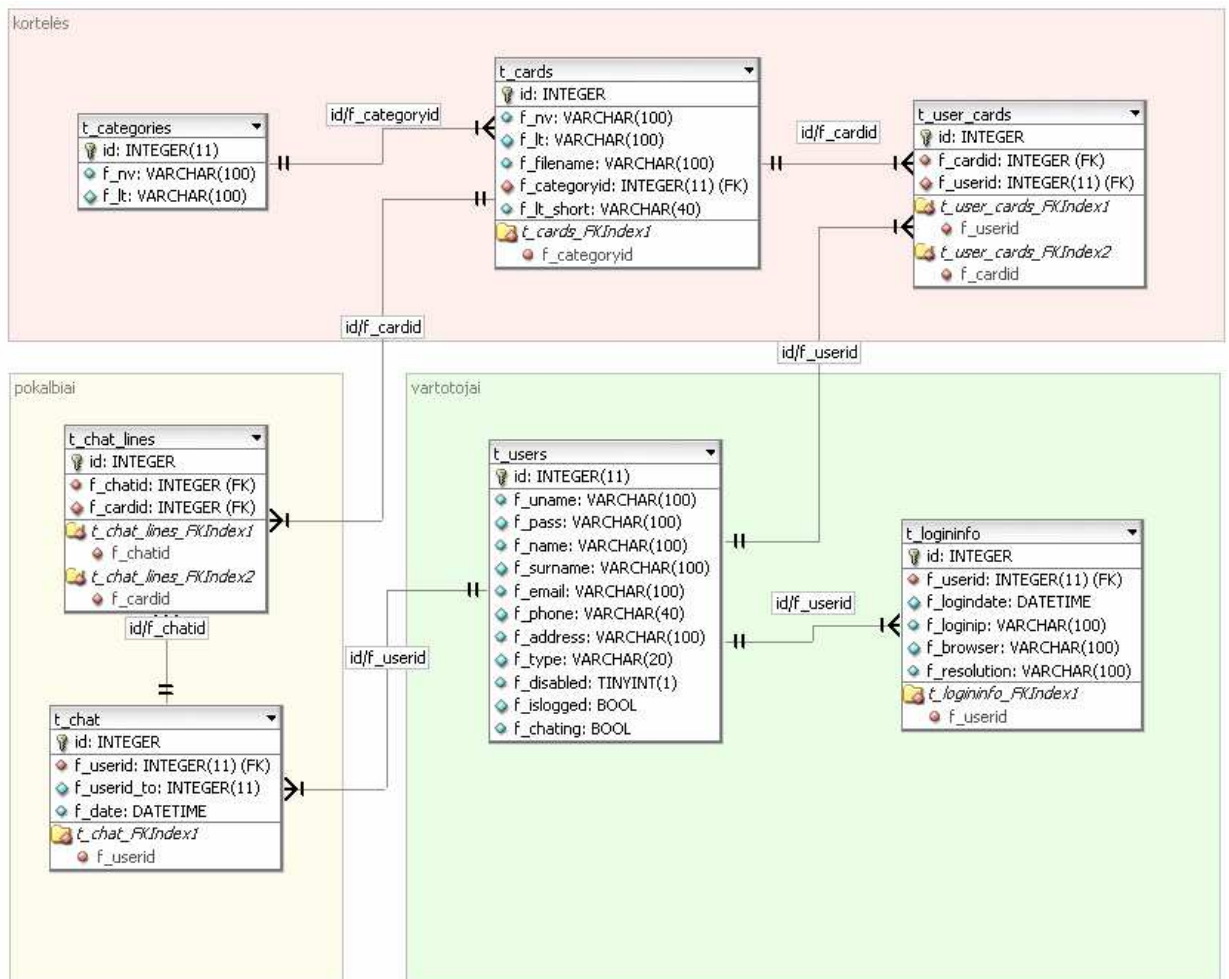
Didelis privalumas naudojant Ajax yra internetinių resursų taupymas. Puslapis kurį reikia perkrauti kas kart kai norima žengti žingsnį naudoją žymiai daugiau resursų negu puslapis kuris naudodamas Ajax užkrauna tik atsinaujinusias puslapio dalis. Tai ypač tinkamas sprendimas šiame darbe aprašomu sistemos kūrimo atveju, kadangi multimedia ir panašaus tipo turinys internete yra ypač nuostolingas kai pagrindinis veikimas paremtas kortelių dėliojimu iš vienos vietos į kitą ir atgal. Ši technologija priklauso Web 2 šeimai (svetainės YouTube, Flickr, Blog ir k.t.).

#### 4.2.7. Duomenų struktūros

Duomenų saugojimui ir jų naudojimui bei kaupimui naudojamas *MySQL 5.0.24a TCP/IP* duomenų bazės serveris. Duomenų bazės administravimui, duomenų įvedimui bei peržiūrai naudojama tokia programinė įranga:

- MySQL Administrator 5.1.11;
- MySQL Query Browser;
- MySQL Migration Toolkit;
- MySQL Workbench;
- phpMyAdmin 2.8.2.4

Šiam projektui sumodeliuota duomenų bazė reikiamiems duomenims, prisijungimui ir pokalbiams realizuoti. Jos schema, regionai, ryšiai ir lentelės pateikiamos žemiau (29 pav.).



29 pav. Duomenų bazės schema

Vizualiai lentelės atskirtos pagal programose naudojamas dalis ir modulius.

Lentelių aprašymai:

t_categories	
id	INTEGER(11)
f_nv	VARCHAR(100)
f_lt	VARCHAR(100)

30 pav. t\_categories struktūra

- *t\_categories* - Kortelių grupėms saugoti skirta lentelė. Priklauso regionui „kortelės“.
  - id – Šiame lauke saugomas lentelės identifikacijos numeris jis kartu yra ir lentelės pirminis raktas, jame įrašai unikalūs;
  - f\_nv – Norvegiška(originali) grupės reikšmė;
  - f\_lt – Lietuviškas kortelių grupės pavadinimas.

t_cards	
id	INTEGER
f_nv	VARCHAR(100)
f_lt	VARCHAR(100)
f_filename	VARCHAR(100)
f_categoryid	INTEGER(11) (FK)
f_lt_short	VARCHAR(40)
t_cards_FKIndex1	
f_categoryid	

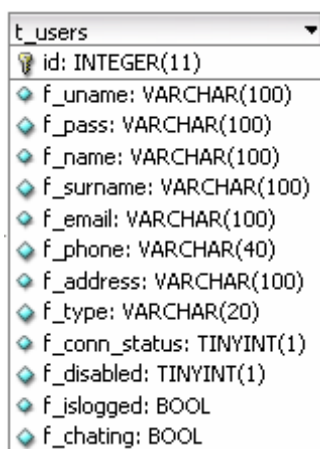
31 pav. t\_cards lentelės struktūra

- *t\_cards* – kortelių pavadinimų, ir buvimo vietos saugojimui skirta lentelė. Priklauso regionui „kortelės“.
  - id – saugomas identifikacinis lentelės numeris ir priminis lentelės raktas;
  - f\_nv – norvegiškas kortelės pavadinimas;
  - f\_lt – lietuviškas kortelės pavadinimas;
  - f\_filename – kortelės failo vardas;
  - f\_categoryid – kortelės jungiamasis laukas (FK), parodantis kuriai kortelių grupei priklauso kortelė;
  - f\_lt\_short – sutrumpintas lietuviškas kortelių pavadinimas;
  - t\_cards\_FKIndex1 – kortelės indeksas.

t_user_cards	
id	INTEGER
f_cardid	INTEGER (FK)
f_userid	INTEGER(11) (FK)
t_user_cards_FKIndex1	
f_userid	
t_user_cards_FKIndex2	
f_cardid	

32 pav. t\_user\_cards lentelės struktūra

- *t\_user\_cards* – vartotojų kortelės. Kortelės kurios surenkamos po kiekvieno pokalbio ir sudedamos kaip dažniausiai naudojamos.
  - id – lentelės pirminis raktas ir identifikacinis laukas, unikalus;
  - f\_cardid – kortelės siejimasis su t\_card.id lauku;
  - f\_userid – kortelės ryšio laukas su t\_user.id. lauku;
  - t\_user\_cards\_FKIndex1 – vartotojo kortelių indeksas su t\_user lentele;
  - t\_user\_cards\_FKIndex2 – vartotojo kortelių indeksas siejantis su t\_cards lentele.



t_users	
id	INTEGER(11)
f_username	VARCHAR(100)
f_pass	VARCHAR(100)
f_name	VARCHAR(100)
f_surname	VARCHAR(100)
f_email	VARCHAR(100)
f_phone	VARCHAR(40)
f_address	VARCHAR(100)
f_type	VARCHAR(20)
f_conn_status	TINYINT(1)
f_disabled	TINYINT(1)
f_islogged	BOOL
f_chating	BOOL

33 pav. t\_users lentelės struktūra

- *t\_users* – visų vartotojų lentelė. Duomenys apie jų prisijungimus ir kontaktinę informaciją, bei atitinkamus vartotojų tipus.
  - id – lentelės pirminis raktas ir identifikacinis laukas, unikalus;
  - f\_username – vartotojo vardas, naudojamas prisijungiant prie sistemos;
  - f\_pass – vartotojo prisijungimo slaptažodis;
  - f\_name – vartotojo tikrasis vardas(nebūtinas laukas);
  - f\_surname – vartotojo pavardė(nebūtinas laukas);
  - f\_email – vartotojo kontaktinis elektroninis paštas(nebūtinas laukas);
  - f\_phone – vartotojo kontaktinis telefonas(nebūtinas laukas);
  - f\_address – vartotojo adresas(nebūtinas laukas);
  - f\_type – identifikacinis tipas kuris skirsto ar tai vartotojas „U“ ar administratorius „A“;
  - f\_conn\_status – nurodoma ar vartotojas prisijungęs prie sistemos ar atsijungęs;
  - f\_disabled – jei administratorius nori blokuoti vartotoją tai vykdoma pakeičiant lauko požymį į „true“ arba į 1;

- `f_islogged` - nurodoma ar vartotojas prisijungęs prie sistemos ar atsijungęs (jei prisijungęs suteikiamas prisijungimo nr.)
- `f_chatting` – nurodoma ar vartotojas jau kalbasi ar ji pasirengęs užmėgsti pokalbį ir galima su juo kalbėtis.

Field Name	Field Type
id	INTEGER
f_userid	INTEGER(11) (FK)
f_logindate	DATETIME
f_loginip	VARCHAR(100)
f_browser	VARCHAR(100)
f_resolution	VARCHAR(100)
Index: t_logininfo_FKIndex1	
f_userid	

34 pav. `t_logininfo` lentelės struktūra

- `t_logininfo` – saugoma ir pildoma informacija apie prisijungusį vartotoją ir jo istoriją (nenumatytiems atvejams). Ši lentelė pildoma kai vartotojas prisijungia prie sistemos.
  - `id` – identifikacinis lentelės numeris, pirminis raktas, unikalus;
  - `f_userid` – prisijungusio vartotojo id. Rišamasis laukas kuris jungiasi su `t_user.id`;
  - `f_logindate` – registruojama paskutinio prisijungimo data;
  - `f_loginip` – registruojamas vartotojo IP;
  - `f_browser` – prisijungusiojo statistika apie interneto naršyklę;
  - `f_resolution` – registruojama informacija apie vartotojo ekrano raišką;
  - `f_logininfo_FKIndex1` – indeksas nurodantis jungimosi su `t_user` lentele pasekmes kai ryšys tarp lentelių panaikinamas. Kai trinamas vartotojas tai informacija iš čia taip pat ištrinama. Kai vartotojo id keičiamas automatiškai pakeičiama rišamojo lauko id.

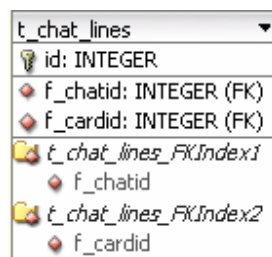
Field Name	Field Type
id	INTEGER
f_userid	INTEGER(11) (FK)
f_userid_to	INTEGER(11)
f_date	DATETIME
Index: t_chat_FKIndex1	
f_userid	

35 pav. `t_chat` lentelės struktūra

- `t_chat` – pokalbio eilučių saugojimas. Priklauso regionui „pokalbiai“.
  - `id` – identifikacinis lentelės numeris, pirminis raktas, unikalus;



- f\_userid – pokalbį pradėjusio vartotojo id. Rišamasis laukas kuris jungiasi su t\_user.id;
- f\_userid\_to – sutikusio kalbėti vartotojo id. Rišamasis laukas kuris jungiasi su t\_user.id;
- f\_date – registruojama paskutinio nusiųsto „žodžio“ data;
- t\_chat\_FKIndex1 – indeksas nurodantis ryšį su t\_chat lentele ir pasekmes kai ryšys panaikinamas. Kai trinamas vartotojas tai informacija iš čia taip pat ištrinama. Kai vartotojo id keičiamas automatiškai pakeičiama rišamojo lauko id.



t_chat_lines	
id	INTEGER
f_chatid	INTEGER (FK)
f_cardid	INTEGER (FK)
t_chat_lines_FKIndex1	f_chatid
t_chat_lines_FKIndex2	f_cardid

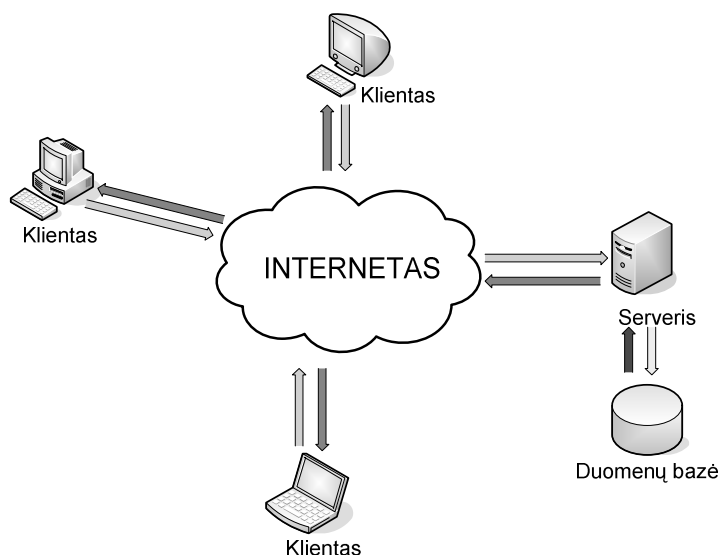
36 pav. t\_chat\_lines lentelės struktura

- t\_chat\_lines – pokalbio, eilučių „žodžių“ saugojimas. Priklauso regionui „pokalbiai“. Pildoma kai vartotojas pasirinko kortelę kurią nori nusiųsti pašnekovui.
  - id – identifikacinis lentelės numeris, pirminis raktas, unikalus;
  - f\_chatid – pokalbio tarp dviejų pašnekovų id. Jis yra jungiamasis laukas kuris jungiasi su t\_chat.id;
  - f\_cardid – rodamos kortelės id. Rišamasis laukas kuris jungiasi su t\_card.id;
  - f\_chat\_lines\_FKIndex1 - indeksas nurodantis ryšį su t\_chat lentele ir pasekmes kai ryšys panaikinamas. Kai trinamas vartotojas, ir jo pokalbiai tai informacija iš čia taip pat ištrinama. Kai pokalbio id keičiamas automatiškai pakeičiama rišamojo lauko id.
  - f\_chat\_lines\_FKIndex2 - indeksas nurodantis ryšį su t\_card lentele ir pasekmes kai ryšys panaikinamas. Kai trinama kortelė tai informacija iš čia taip pat ištrinama. Kai kortelės id keičiamas automatiškai pakeičiama rišamojo lauko id.

Pageidaujant duomenų bazę ateityje galima papildyti naujomis lentelėmis, laukais ir ryšiais, nes nuo to IS veikimas nepriklauso.

### 4.3. Bendradarbiaujančios sistemos

Projekte tarpusavyje bendradarbiauja kelios sistemos drauge. Serverio sistema (Apache) kuri atlieka pagrindinį vaidmenį įgyvendinant komunikaciją tarp vartotojo kompiuterio ir nutolusio serverio. Jos veikimo principas pavaizduotas žemiau(37 pav.).



37 pav. Kliento - serverio architektūra

Serveryje taip pat naudojama duomenų bazė kuri taip pat veikia drauge su serveriu kaip neatskiriama serverio ir šio projekto dalis. Serveris ir duomenų bazė turi tarpusavio ryšį kurio metu kai vartotojas nusiunčia komandą serveriui, serveris reaguoja ir jei reikalinga duomenų bazė jis kreipiasi į duomenų bazę prašydamas užklauskos kurią suformulavo vartotojas. Gavus atsakymą iš duomenų bazės serveris jį persiunčia vartotojui kuris užklauską siuntė.

Šiame projekte įtrauktas ne tik nuotolinis serveris bet ir pats vartotojo PK. Jis naudojamas pranešimams, veiksams ir turiniui formuoti bei valdyti. Puslapis sukurtas taip, kad sumažinti vartotojo puslapio perkrovimo skaičių.

### 4.4. Rizikos įvertinimas

Rizikos identifikavimas. Galimi rizikos tipai:

- Technologinė rizika. Pavyzdžiui, pasikeičia duomenų bazės valdymo sistemos versija ir atsiranda papildomas konfigūravimas sistemos derinimas arba net visos sukurtos sistemos perdarymas;

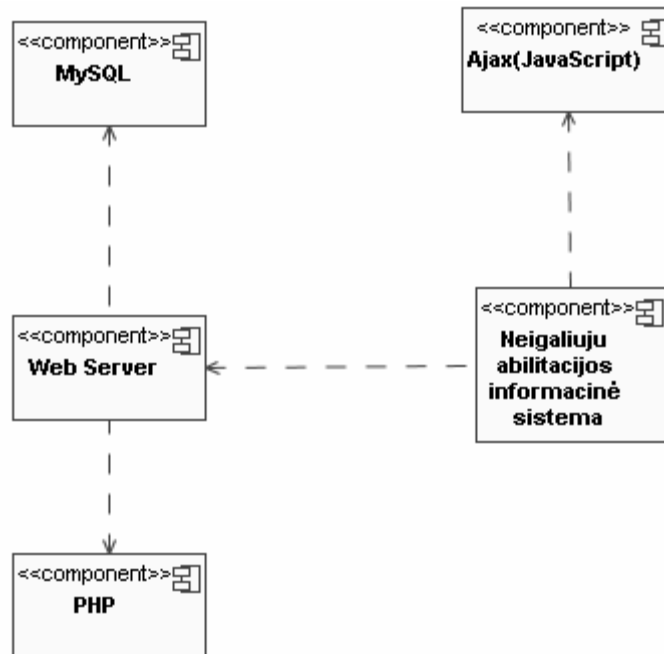
- Žmonių rizika. Pavyzdžiui, komandos nario liga, ar pasitraukimas iš komandos. Taip pat paskirtos užduoties neįgyvendinimas, dėl per didelio užsibrėžto tikslo ir dėl laiko stokos;
- Įrankių rizika. Keičiasi įrankių naudojamoje sistemoje versijos ir funkcionalumas. Pasirenkamas neefektyvus ar reikalingų savybių neturintis darbo įrankis kuris nesuteikia tiek galimybių kiek suteiktų kitas tokio pobūdžio įrankis;
- Reikalavimų rizika. Kai, pasikeičia reikalavimai sistemos funkcionalumui. Į šią rizikos grupę įeina ir neplanuotų reikalavimų identifikavimas (reikalavimai kurie užsakovui buvo „savaiame suprantami“);
- Įvertinimo rizika. Pavyzdžiui, per trumpas projekto realizavimo laikas, blogai įvertintas PĮ dydis, suplanuotas per mažas galimų problemų kiekis.

Sprendimai susidūrus su išvardintais rizikos faktoriais arba būdai rizikai išvengti:

- Kaip galima labiau numatyti visas technologines rizikas ir kuriant sistemą pasiruošti, kad pasikeitimų įgyvendinimas pareikalautų kuo mažiau resursų (darbo, laiko ir t.t.)
- Žmonių rizikos sumažinimui turi įtakos bendradarbiavimas tarp projekto komandos narių. Kiekvienas žmogus dirbantis prie sistemos turės bendrą supratimą kokioje stadijoje sistemos kūrimas yra kiekvienu laiko momentu. Toks žinojimas bus pasiektas periodiškais pasitarimais, ataskaitomis, diskusijomis. Bus vengiama tokio darbo pobūdžio, kai darbuotojas žino tik apie tai ką jis daro, nesuvokdamas ką daro kiti. Tokiu būdu atsiras galimybė dirbti porose, pavaduoti susirgusį ar išėjusį komandos narį.
- Kad sumažinti šią riziką įrankiai bus renkami ypač atsakingai. Visados bus turimi keli atsarginiai variantai.
- Bus stengtasi išsiaiškinti reikalavimus kaip galima tiksliau ir pilniau. Nuo reikalavimų pasikeitimo bus saugomasi, modeliuojant būsimą sistemą kartu su užsakovu ir tokiu būdu bandyti numatyti tuos būsimus pasikeitimus ir atlikti juos dar prieš pradėdant sistemos kūrimą.

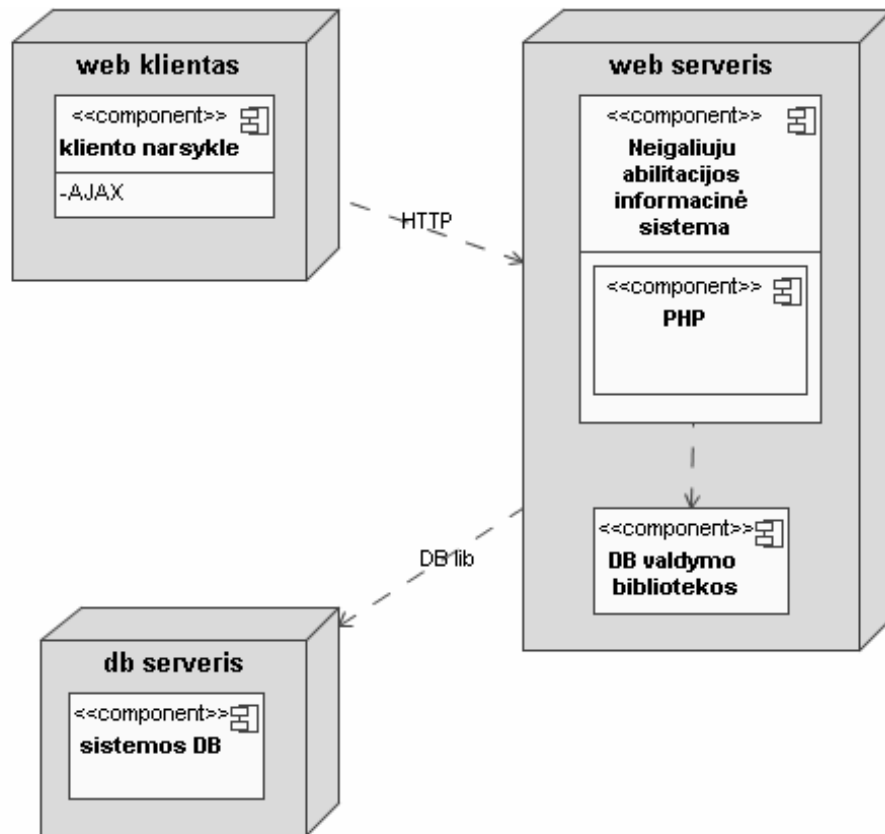
#### 4.5. Sistemos komponentai ir įdiegimas

Komponentų diagrama rodo fizinį sistemos vaizdą: komponentus bei jų tarpusavio priklausomybes. Sistemos pagrindiniai komponentai pateikiami žemiau (38 pav.)



38 pav. Sistemos komponentų diagrama

Įdiegimo diagramose atvaizduojami procesoriai, kurių pagalba sistema veikia. Ši diagrama vaizduojama žemesniame paveiksle(39 pav.)



39 pav. Sistemos įdiegimo diagrama

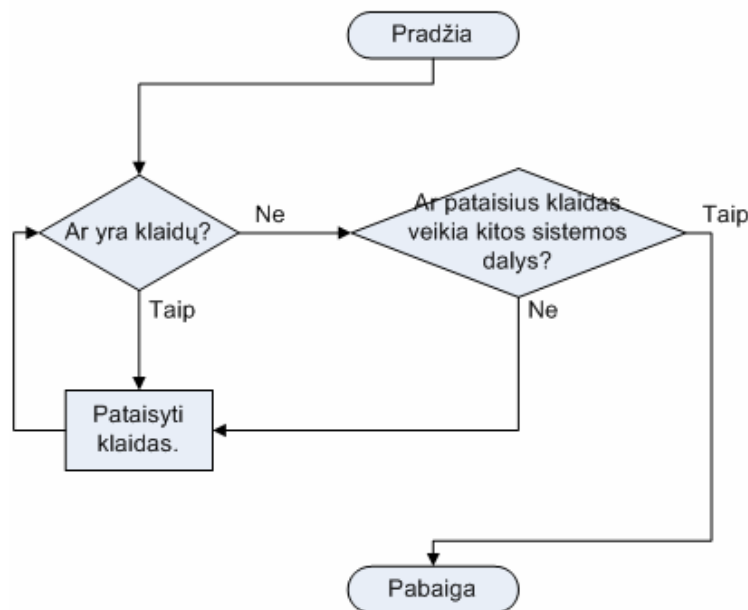
## 5. Testavimas

### 5.1. Testavimo etapai

Testavimo etapai suskirstyti į dvi dalis. Jos glaudžiai susijusios viena su kita, nes vienai iš jų palikus neveiksniai kita dalis ir visa sistema praranda savo funkcionalumą ir stabilų veikimą.

Testavimo etapai:

Programinės dalies testavimas. Programinės dalies testavimui buvo panaudotos visos susijusios programos, visos susijusios operacinės sistemos komandinės dalys ir susijusios duomenų bazės dalys. Testavimo metu iškilusios problemos, kurių buvo labai daug, nes tai nauja sistema, buvo šalinamos tam tikrais, šiai problemai panaikinti, skirtais metodais. Kiekvieno testavimo ir sutvarkymo metu buvo užrašomi testavimo duomenys ir kas pakeista bei kaip tai įtakojo sistemos darbą. Po tokio testavimo viskas kartojama iš naujo ir taisomos tos pačios arba naujai atsiradusios problemos. Kartojama tol kol kiekviena dalis veikia ir nesukelia konfliktinių situacijų su jau anksčiau veikusiomis kitomis dalimis. Testavimo algoritmas pavaizduotas žemiau (40 pav.)



40 pav. Testavimo algoritmas

Šis algoritmas buvo pritaikytas daugeliui programinei pusei priklausančių dalių ir sprendžiant iš galutinio sistemos veikimo jis pasiteisino.

- *Funkcinis testavimas* – Šio testavimo metu programiniame kode buvo testuojama sistemai priklausančios funkcijos, metodai, klasės, susijungimai pasiekiamumas ir visos susijusios vartotojo sąryšio (angl. user interface) dalys. Testavimo algoritmas labai panašus į (pav. 7) pavaizduotą testavimo algoritmą, išskyrus tai, kad šis metodas buvo pritaikomas atskiroms

sistemos dalims. Šios dalys susmulkintos į dar mažesnes dalis, o pastarosios jei reikia į dar mažesnes dalis. Taip gaunamas panašios į medžios struktūrą, testavimo dalys. Išaiškėjus, kad neveikia kuri nors viena iš dalių, ji pataisoma ir taip išvengiama taisymo klaidų atsikartojimo kitose dalyse.

Testuojamos sistemos dalys:

- Serveris
  - Serverio konfigūracija
  - Apache versijų suderinamumas
    - PHP palaikymas
    - PHP versijos tikrinimas
    - GL lib palaikymas(paveikslėlių apdorojimui)
- Duomenų bazė
  - Versijų suderinamumas
  - Reikiamo funkcionalumo tikrinimas
  - Racionalumo tikrinimas
- Registracijos langas
  - Pašalinių asmenų draudimas
  - Tikėtinas veikimas
  - Duomenų saugojimas
  - Duomenų trynimas
  - Duomenų redagavimas
- Prisijungimo langas
  - Pašalinių asmenų draudimas
  - Teisingas prijungimas
  - Teisingas autorizavimas
    - Teisingas autentifikavimas
- Pagrindinis langas
  - Tikėtinas veikimas
  - Meniu
  - Įvedimo įrenginių funkcionavimas
    - Klaviatūros veikimas
    - Pelės veikimas
  - Pagrindinių kortelių veikimas

- Pokalbių juostos veikimas
- Istorijos veikimas
- Statistikos veikimas

Šis testavimo medis buvo skirstomas į dar smulkesnes dalis jei vienoje iš šakų įvykdavo klaida ir negrįžtama tol kol toje šakoje viskas funkcionuoja kaip ir buvo laukta.

## 5.2. Testavimo planas ir rezultatai

Testavimo modelyje atsispindi sistemos testavimo veiksmų seka bei testavimo duomenys ir gauti sistemos rezultatai.

5 lentelė. Prisijungimo testavimas

Veiksmas	Sistemos reakcija
1. Įvedamas neegzistuojantis prisijungimo vardas (arba neįvedamas joks) <i>Pvz. „“</i>	Pranešimas. Prisijungimo operacija sustabdoma. Siūloma registruotis svetainėje kaip naujas vartotojas.
2. Įvedamas netinkamas prisijungimo slaptažodis <i>Pvz. „aasdas“</i>	Pranešimas. Prisijungimo operacija sustabdoma. Siūloma pakartoti dar kartą.

6 lentelė. Slaptažodžio keitimo testavimas

Veiksmas	Sistemos reakcija
3. Įvedamas neegzistuojantis prisijungimo vardas (arba neįvedamas joks)	Pranešimas. Pakeitimo operacija sustabdoma. Veiksmo pakartojimas.
4. Įvedamas netinkamas prisijungimo slaptažodis	Pranešimas. Pakeitimas sustabdomas
5. Naujas slaptažodis pakartojus du kartus nesutampa <i>Pvz. pirmas „123“, antras „321“</i>	Pranešimas. Pakeitimas sustabdomas
6. Naujame slaptažodyje <=1 simbolis. <i>Pvz. „h“</i>	Pranešimas. Pakeitimas sustabdomas

7 lentelė. Virtualios klaviatūros testavimas

Veiksmas	Sistemos reakcija
7. Paspaudžiamas virtualios klaviatūros iškvietimo mygtukas	Parodoma klaviatūra
8. Virtualios klaviatūros klavišų testavimas.	Sistema funkcionuoja kaip ir elgiantis su tikra klaviatūra.
9. Bandyamas sutrikdyti klaviatūros darbą <i>Pvz. Spaudinėjami visi mygtukai iš eilės</i>	Sistema funkcionuoja kaip ir elgiantis su tikra klaviatūra. Neįvyksta joks išskirtinis sistemos pasikeitimas

8 lentelė. Dinaminio prisijungimo meniu testavimas

Veiksmas	Sistemos reakcija
10. Spaudžiama prisijungimo vardo raidė	Sistema užkrauna visus vartotojų vardus su šia raide ir parodo centriniame iškrentančiame meniu
11. Spaudžiamas iškrentančio dialogo iškvietimo mygtukas	Iškviečiamas dialogas su prieš tai užkrautomis vartotojų vardų ir pavardžių sąrašu.
12. Neparenkama iškrentančio dialogo reikšmė	Parodomas pranešimas. Sistema nebedirba toliau.



9 pav. Pagrindinis sistemos langas

<b>Veiksmas</b>	<b>Sistemos reakcija</b>
13. Ar užkraunamas sisteminis langas	Sisteminis langas užkraunamas sėkmingai su Veiklos grupės kortelėmis
14. Spaudžiamas pele grupių pasirinkimo mygtukas	Pasirinkimo gupių mygtukas funkcionuoja nepriekaištingai. Išvedamos visos grupės ir jų pavadinimai.
15. Spaudžiamas pele „Mano“ kortelių mygtukas	Mygtukas parodo visas nesena ir dažniausiai naudotas korteles.
16. Testuojami kontroliniai svetainės klavišai Z,X,C,V,B,N	Valdymo klavišai veikia teisingai. Pažymi atitinkamą eilutę kortelių laukeliuose, o paspaudus dar kartą inicijuojamos kitos eilutės pažymėjimo komanda. Kiekvienas laukas prieš pažymint išvalomas. Meniu iškviečiamieji klavišai taip pat veikia teisingai – parodo iškrentančius meniu.
17. Testuojami mygtukai 0 – 9	Veikimas kaip ir tikėtasi veikia nepriekaištingai. Pažymimi visi kiekvieno skaičiaus atskirai, kortelių laukų stulpeliai, o paspaudus antrą kartą kortelės stulpelis pasikeičia į sekantį.
18. Testuojamas kortelės įdėjimas į pokalbio eilę	Kortelė įdedama į pokalbio eilę ir atsiranda ją žymintis tekstas aukščiau jos.
19. Testuojamas kortelės išėmimas iš pokalbio eilės klaviatūra	Paskutinė įdėta kortelė išimama iš pokalbių eilės ir vaizduojamos tik paskutiniosios pokalbio eilėje esančios kortelės jei kortelių nėra nieko neįvyksta.
20. Kortelės išėmimas naudojantis pelės paspaudimais	Pelės paspaudimo metu iš pokalbio sąrašo išnyksta ta kortelė ant kurios buvo spaudžiama.
21. Kortelės padidinimas užvedus ant kortelių lauko	Kortelė padidėja ir parodoma jos grupė ir pavadinimas.
22. Kortelės padidinimas užvedus ant kortelių pokalbių lauko	Kortelė padidėja ir parodoma jos grupė ir pavadinimas.
23. Klavišų „<“ ir „>“ testavimas ant didesnio kortelių kiekio	Kai spaudžiami klavišai „<“ ir „>“ svetainės kortelių turinys keičiasi
24. Klavišų kombinacijos „skaičius“ – „raidė“ testavimas	Testavimo esmė ištestuoti ar veikia dviejų klavišų kombinacijos, kai spaudžiamos visos galimos kontrolinės raidės, o netrukus ir skaičiai kombinacijos. Rezultate gauta, jog susikirtimo taške esanti kortelė įkrenta į pokalbių eilutę ko ir tikėtasi iš sistemos.
25. Klavišų kombinacijos „raidė“ – „skaičius“ testavimas	Testavimo esmė ištestuoti ar veikia dviejų klavišų kombinacijos, kai spaudžiami visi galimi kontroliniai skaičiai, o netrukus ir raidės - kombinacijos. Rezultate gauta, jog susikirtimo taške esanti kortelė įkrenta į pokalbių eilutę ko ir tikėtasi iš sistemos.
26. Atsijungimo testavimas	Pasirinkus išjungimo mygtuką sugrįžtama į prisijungimo langą užsaugant vartotojo sesijos informaciją duomenų bazėje.

### *5.3. Išvados*

- Testavimas ir testavimo rezultatai padėjo suprasti kokios galimos iškylančios problemos, ir kaip efektyviau jas spręsti kur reikia padirbėti labiau, kad tokių klaidingų veikimo situacijų būtų išvengta.
- Išsamiai ištestavus sukurtą produktą buvo įvertinta testavimo svarba ir testavimo būtinumas. Netestuojant naujai sukurto produkto negalima tikėtis nei gerų rezultatų nei produkto korektiško veikimo.
- Testavimo metu buvo testuojama integravimas į kitas sistemas kuris parodė, jog nedideli pakeitimai sistemos kode kurie gali duoti dar platesnį sistemos pritaikymo spektrą.

## Išvados

1. Šio darbo metu buvo išnagrinėtos įvairios neįgaliesiems sukurtos IS, žaidimai, mokymo priemonės, pagalbinės priemonės, jų vienokiems ar kitokiems bendravimo įgūdžiams lavinti ir skatinti susidomėjimą. Analizės rezultatai leido savo kuriamai sistemai atrinkti svarbius bendravimo kompiuteriu aspektus, tokius kaip patogumas, naudingumas, interaktyvumas. Taip pat sužinoti neįgaliųjų poreikius, ir neįgaliųjų bendravimo su kompiuteriu principus, galimybes, siekiamus tikslus ir prognozuojamus rezultatus.
2. Buvo išanalizuoti šių sistemų pagrindiniai trūkumai ir privalumai išdėstyti argumentai, kodėl vieni būdai yra geresni ar blogesni už kitus, Išrinkti kurie metodai labiau atitinka šioje sistemoje keliamus tikslus ir uždavinius. Išnagrinėti pagrindiniai neįgaliųjų poreikiai ir siekiai naudojant tokias sistemas, įvertintas IS naudingumas ir tokios sistemos būtinumas. Be to šio darbo metu buvo analizuojamos egzistuojantys neįgaliesiems skirti įvedimo įrenginiai, t.y. specialios kompiuterinės pelės, klaviatūros, valdymo paneliai, planšetės, šviesos diodai, infraraudonais spinduliais paremti įrenginiai. Išanalizuotos papildomos klaviatūros ar kitokios kompiuterio valdymo priemonės skirtos neįgaliesiems ir pritaikomos šio projekto darbe, tačiau pastebėta, jog Lietuvos rinkoje tai dar nėra aktuolu.
3. Buvo sukurtas primityvus prototipas-demonstracinė programos versija pritaikyta darbui su neįgaliaisiais, o vėliau ir veikiantis galutinis programos modelis, kurios pagrindinis privalumas tai, kad jis sėkmingai gali būti valdomas tiek pasinaudojant kompiuterine pele tiek ir klaviatūra, t.y., nepriklausomai nuo žmogaus sugebėjimo adaptuotis prie kompiuterio sistemos. Rasti sprendimai kaip pateikti korteles suprantama ir vaikams paprastesne forma, panaudojant elektrinius sprendimus. Sistema buvo kuriama tam, kad būtų panaudota kaip informacinė mokomoji, bendravimą kortelėmis palengvinanti priemonė skirta neįgaliesiems, bet tinkanti ir sveikiems vaikams.
4. Sugalvotas ir pritaikytas būdas Norvegų naudojamas bendravimo ideogramas perkelti iš stacionarios būsenos(popierinės) į dinaminę. Sėkmingai pritaikytos naujos AJAX technologijos leidžiančios maksimaliai išnaudoti naujas galimybes šiame projekte. Projektas pasiekiamas internetu tam, kad kiekvienas kompiuteriu sugebantis naudotis neįgalusis galėtų šios sistemos pagalba bendrauti, specialiomis kortelėmis, tarpusavyje su savo auklėtojomis, bendraamžiais, draugais, tėvais ar su kitais vaikais.

5. Pastebėta, jog tokių neįgaliųjų bendravimą skatinančių sistemų nėra daug ir tai nėra populiaru, o ypač nėra tokios srities pigių tuo labiau nemokamų produktų. Lietuvoje ši sritis paliesta visai menkai todėl projekto kūrimo metu buvo susiduriama su informacijos apie tokias sistemas trūkumas. Darbo metu taip pat buvo susidurta su kompiuterinės įrangos skirtos neįgaliesiems stoka. Pabrėžtina, jog įmonių užsiimančių kompiuterinės įrangos platinimu Lietuvoje labai nedaug, o ir tokių įrenginių kaina atitinkamai daug didesnė nei įprastinių komponentų.
6. Sukurta programa ir jos konstrukcija demonstruoja pagrindines naujų technologijų panaudojimo galimybes nestandartinėse situacijose. Valdymo per web sąsają principus ir tokio valdymo lankstumo, galimybes, taip pat pabrėžiamos plačios ir neišnaudotas galimybes šioje specifinėje informacinių sistemų srityje. Atveriamos naujos vizijos į kitoki – specializuotą šiuolaikinės kompiuterijos pritaikymą žmonijos gerovei ir visuomeninei naudai.
7. Programos galutinei versijai realizuoti buvo pasirinktos šios kūrimo sistemos:
  - XHTML – šabloninio veikimo realizavimui skirta žymėjimo kalba, pasižyminti naujais sugriežtintais standartais kurie veda prie naršyklių vienodinimo;
  - PHP – pagrindinė projekte naudota internetinio programavimo kalba kuri leidžianti kartu apjungti visas sistemos dalis ir kūrimo priemones į vieną bendrą visumą. Pagrindiniai programavimo kalbos privalumai, jog palaikomas objektinis programavimas ir išlaikoma C++ sintaksė ;
  - MySQL – naujausia (5 versija) duomenų bazės valdymo ir duomenų saugojimo sistema palaikanti SQL-92 standartus;
  - AJAX – (asinchroninis JavaScript ir XML). Tai technologijų rinkinys leidžia neperkraunant puslapio atnaujinti tam tikras reikiamas dalis ir reaguoti į vartotojo poreikių pasikeitimus svetainėje. Pasirinkimo priežastys: taupomas vartotojo laikas ir taupomi interneto resursai.
8. Sukurta programos versija viešai prieigai sukonfigūruota ir patalpinta interneto svetainėje [31].

## Terminų ir santraukų žodynas

AAK - Augmentatyvioji ar Alternatyvioji Komunikacija

AJAX (Asynchronous JavaScript and XML) – Asinchroninė JavaScript kalba ir XML

AK – asmeninis kompiuteris

CBT (Computer-based Training) – Kompiuterinis lavinimas.

CSS (Cascading Style Sheet) – html stilių failų formatas

DOM (Document Object Model) – Objektinis dokumento modelis

GUI – (Graphical user interface) – grafinė vartotojo sąsaja

HTML (HyperText Markup Language) - Hiperteksto žymėjimo kalba.

IS – Informacinė sistema

IE – (Internet Explorer) – Populiari Microsoft Interneto naršyklė

FF – (FireFox) – Mozilla interneto naršyklė

OS – Operacinė sistema

TK – tinklo kompiuteris

UML – unifikauta modeliavimo kalba (Unified Modeling Language)

XML (eXtended Mark up Language) - „žymėjimo“ kalba dokumentams.

W3C (World Wide Web Consortium) – Pasaulinis interneto puslapių konsorciumas

KVAC – Kauno Vaikų Abilitacijos Centras

## Literatūros sąrašas

### **Knygos**

1. *Danny Goodman, JavaScript & DHTML Cookbook, Oreily, 2003, -540p.*
2. *Julie C Meloni, Sams Teach Yourself PHP, MySQL™ and Apache in 24 Hours, , Sams Publishing 2002, -528p.*
3. *Julie C Meloni, PHP Fast & Easy Web Development, 2nd Edition, Premier Press, 2002, -481p.*
4. *William J. Pard, XML in Action Web technology, Microsoft Press, 1999, 300p,*
5. *Paul DuBois, MySQL Second Edition. – V.: Sams, 2003. – 1248p.*
6. *Thomas A. Powell , HTML & XHTML: The Complete Reference, Fourth Edition. –V.: McGraw-Hill, 2003. – 932p.*
7. *Christopher Schmitt, CSS Cookbook. –V.: O'Reilly, 2004. -272p.*
8. *Eric A. Meyer, More Eric Meyer on CSS. –V.: New Riders Publishing, 2004. – 304p.*
9. *Danny Goodman, JavaScript & DHTML Cookbook. – V.: O'Reilly, 2003. – 540p.*
10. *Mark Maslakowski, Teach Yourself MySQL in 21 Days. –V.: Sams, 2000. -532p.*
11. *David Axmark, Michael (Monty) Widenius, Jeremy Cole, Arjen Lentz, and Paul DuBois. MySQL Technical Reference. –V.: MySQL AB 2002. -833p.*
12. *Richard York, CSS Instant Results –V: Wrox Press, 2006. -408p.*
13. *David Sawyer McFarland CSS: The Missing Manual –V: O'Reilly, 2006. -494p.*
14. *H . Stephen Kaye, Ph.D., Computer and Internet Use Among People with Disabilities –V: Disability Statistics Center Institute for Health and Aging University of California San Francisco, California 2000. -17p.*

### **Internetas**

15. *The phpMyAdmin Project. 2006 rugsėjis. [žiūrėta 2006-09-29]. Priega per Internetą: [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php). [angl.]*
16. *Apie Lovaas ligą. 2006 sausis. [žiūrėta 2005-11-29]. Priega per Internetą: <http://autism.about.com/cs/behaviorialissues/a/lovaas.htm> [angl.]*
17. *Lietuvių kalba informacinėse technologijose. 2004 vasaris. [žiūrėta 2006-02-20]. Priega per Internetą: [http://www.likit.lt/frames/turinys/tur\\_d.htm](http://www.likit.lt/frames/turinys/tur_d.htm) - [liet.]*
18. *Kauno Vaikų Abilitacijos Centras trumpa informacija . 2003 gegužė. [žiūrėta 2006-02-20]. Priega per Internetą: <http://autizmo.projektas.lt/KVAC.html> - [liet.]*

19. Mayer Johnson programinės įrangos svetainė neįgaliesiems. 2004 birželis. [žiūrėta 2006-03.13] Prieiga per Internetą: <http://www.mayer-johnson.com/> - [angl.]
20. Widgit Home Page mokomosios programinės įrangos neįgaliesiems svetainė. 2006 kovas. [žiūrėta 2006-03.14] Prieiga per Internetą: <http://www.widgit.com/> - [angl.]
21. The Oxford ACE Centre – Oksfordo centras neįgaliesiems. 2006 kovas. [žiūrėta 2006-03.14] Prieiga per Internetą: <http://ace-centre.hostinguk.com/> - [angl.]
22. Kitokį vaiką mylėti sunkiau. 2006 rugsėjis. [žiūrėta 2006.09.30] Prieiga per Internetą: [http://www.lsveikata.lt/index.php?page\\_id=13&news\\_id=1509](http://www.lsveikata.lt/index.php?page_id=13&news_id=1509) –[liet.]
23. Ajax – SWIK. 2006 balandis. [žiūrėta 2006.07.20] Prieiga per Internetą: <http://swik.net/Ajax> - [angl.]
24. MooTools. 2006 kovas. [žiūrėta 2006.07.20] Prieiga per Internetą: <http://mootools.net/> - [angl.]
25. DHTML Tools. 2006 sausis. [žiūrėta 2006.07.20] Prieiga per Internetą: <http://www.dhtmlgoodies.com/> - [angl.]
26. Browser statistics. 2006 rugpjūtis. [žiūrėta 2006.09.10] Prieiga per Internetą: [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp) - [angl.]
27. Netcraft: November 2006 Web Server Survey. 2006 lapkritis. [žiūrėta 2006.11.11] Prieiga per Internetą: [http://news.netcraft.com/archives/2006/11/01/november\\_2006\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2006/11/01/november_2006_web_server_survey.html)
28. PHP: PHP Usage Stats. 2006 Spalis. [žiūrėta 2006.10.09] Prieiga per Internetą: <http://www.php.net/usage.php> - [angl.]
29. Top Ten Reasons To Use MySQL. 2006 Spalis [žiūrėta 2006.09.24] Prieiga per Internetą: <http://www.mysql.com/why-mysql/toptenreasons.html> - [angl.]
30. Ajax: A New Approach to Web Applications. 2005 Vasaris [žiūrėta 2006.07.20] Prieiga per Internetą: <http://www.adaptivepath.com/publications/essays/archives/000385.php> - [angl.]
31. Vaikų abilitacijos svetainė. 2006 gruodis. [žiūrėta: 2007.01.03]. Prieiga per internetą: <http://isd.ktu.lt/mag06/darius/> - [liet.]

Priedai



Priedas Nr. 1

## Magistrinio darbo informacija ir programinė įranga

Čia pateikiami IS priedai kurie reikalingi sukurtos programos paleidimui, diegimui, testavimui, peržiūrai, tolesniam vystymui. Instaliavimo instrukcijas galima rasti šiame dokumente bei kompaktinėje plokštelėje.

Kompaktinės plokštelės sudėtis:

- šio projekto failai(dokumentai, surinkta medžiaga ir );
- šio projekto, internetu paleidžiama programa neįgaliesiems („vas“ katalogas);
- aprašymas kaip instaliuoti sistemą („readme.txt“ failas šakniniame kataloge)
- XAMPP for Windows – Apache 2 http serveris palaikantis ;
- phpMyAdmin – duomenų bazių prisijungimo ir valdymo programa (paleidžiama iš XAMPP katalogo);
- PSPad – nemokamas, galingas programos „Notepad“ pakaitalas - redaktorius;



Vieta kompaktinei plokštelei

Priedas Nr. 2

# PROGRAMUOTOJO VADOVAS

## 1. Pristatymas

Programa skirta neįgaliesiems padėti bendrauti korteline sistema(ideogramomis). Palengvinti kai kuriuos bendravimo aspektus naudojant kompiuteriją ir Ajax technologijas.

Programos veikimas paremtas PHP, JavaScript, AJAX, MySQL ir kitomis web programavimo technologijomis. Norint suprasti kaip veikia ši sistema būtina patirtis susidūrus su visomis išvardintomis technologijomis arba bent turėti analogiškos web programavimo patirties.

Šis projektas yra lengvai atnaujinamas todėl šis programuotojo vadovas turėtų užvesti ant reikiamo kelio norint tęsti atnaujinimo darbus.

## 2. Įdiegimas

Visi programos failai saugomi kompaktinėje plokštelėje kataloge „vas“. Tai yra šakninis programos katalogas. Toliau bus kalbama įvertinant, jog skaitytojas žino kur yra šakninis katalogas.

Sistema gali būti įdiegta į Linux arba Windows serverius. Operacinėje sistemoje turi veikti web serveris su PHP 4+ versija ir MySQL 4.1+ versijos duomenų baze (galima naudoti alternatyvią duomenų bazę tačiau čia tai nebus aprašoma).

Paprasčiausias įdiegimo būdas perkėti kompaktinėje plokštelėje šakniniame kataloge esančius failus ir pakatalogius į pasirinktą web serverį (Windows ar linux). Serverio operacinės sistemos įtakos programai neturi.

Reikalingas naujos duomenų bazės sukūrimas. Domenų bazės pavadinimas naudojamas vėliau konfigūraciniame faile. Sukūrus duomenų bazę su vartotoju kuris turi teisę duomenų bazėje kurti ir trinti lenteles įvykdomas šakniniame kataloge gulintis skriptas „DB\_struct\_data.sql“ failas. Skripto vykdymo metu neturi iškilti jokių klaidų ar pranešimų. Klaidos atveju pravartu pasitikslinti duomenų bazės versiją ir paskaityti gamintojo nurodymus.

**Dėmesio:** šio failo keitimas gali įtakoti duomenų korektiškumą perkėlus į duomenų bazę. Sėkmingai sukūrus lenteles duomenų bazėje turi būti sukuriamas vartotojas kurio „f\_type“ – 1 (Administratorius).

Atlikus šiuos veiksmus ir surinkus atitinkamą serverio adresą interneto naršyklėje turime pamatyti pagrindinį prisijungimo langą. Lango viršuje juodu šriftu matomi klaidos pranešimai kurie byloja apie nepilnai sukonfigūruotą sistemą. Kaip sukonfigūruoti aprašoma žemiau skyrelyje „konfigūravimas“.

### 3. Konfigūravimas

Konfigūravimas atliekamas šakniniame kataloge gulinčiame faile config.php. Jame turi būti nurodytas prisijungimo vardas, slaptažodis, duomenų bazė ir duomenų bazės serveris. Šio failo pavyzdys su užpildytai išvardintai duomenų bazės duomenimis pateikimas žemiau.

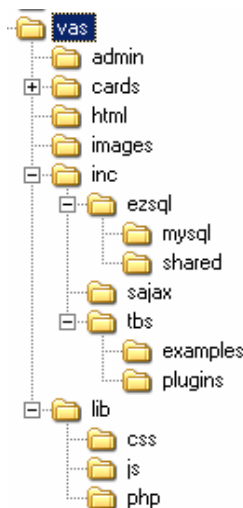
```
1 <?php
2
3 $db['uname'] = "vas";
4 $db['passw'] = "slaptazodis12";
5 $db['db'] = "vas_db";
6 $db['host'] = "localhost";
7
8 ?>
```

Teisingai užpildžius ir sukonfigūravus bei išsaugojus failą galima atnaujinti titulinį puslapį ir jame be klaidų turėtų būti rodomas prisijungimo puslapio vaizdas. Tai byloja apie sėkmingą programos paleidimą. Taip pat galima pateisti „testdb.php“ failą ir įsitikinti programos veikimo bei sukonfigūravimo teisingumu. Paleidus šį failą turi matytis duomenų bazės versija.

**Pastaba:** Kompaktinėje plokštelėje pateikiama programinė įranga padėsianti lengvai įdiegti web serverio sistemą Windows XP ir aukštesnėje aplinkoje į programuotojo kompiuterį. Daugiau informacijos galima sužinoti ir kaip įdiegti šį serverį į kitas operacines sistemas XAMPP.org svetainėje.

### 4. Failų struktūra ir sąsajos

Projekto katalogai išdėstyti šia tvarka ir kitokia failų pavadinimų ar jų išdėstymo tvarkos keitimas gali įtakoti tolesnę programos veikimo schemą. Visi failai ir katalogai saugomi standartinėmis „read-write“ teisėmis. Tokios teisės turėtų ir išlikti norint pasiekti pageidaujamą rezultatą. Failų struktūra vaizduojama žemiau esančiame paveiksle.



Katalogų aprašymai:

**Admin** – kataloge saugomas svetainės administratoriaus prisijungimo ir administravimo portalas. Administratorius jungiasi atskirai nuo visų kitų registruotų vartotojų. Pagrindinis paleidimo failas „admin.php“ arba tiesiog „index.php“ kuris savo ruožtu iškviečia failą „admin.php“. Jis naudoja šablono admin.html karkasą.

**Cards** – katalogas skirtas vien tik vaizdinei informacijai. Čia saugomos visos kortelės skirtos darbui su sistema. Esant reikalui galimas kortelių papildymas. Kataloge kortelės sugrupuotos pagal abėcėlę. Abėcėlė nusako kokias grupes, kokios kortelės priklauso, pvz. A- „žmonės“. Tai galima sužinoti iš duomenų bazės įrašų, lentelėje t\_categories, f\_letter – raidė, f\_lt – grupės atitikmuo. Be to kortelės saugomos 4 skirtingais dydžiais – programos veikimo internetu pagreitinimui. Kiekvienas kortelių dydis identifikuojamas pridėdam gale atitinkamai po galūnę:

- Nieko. Tai originali kortelė paruošta apdorojimui – sumažinimui iki reikiamo dydžio.
- „\_b“ nuo angliško žodžio „big“ – didžiausios sistemoje naudojamos kortelės (200px x 200px). Naudojamos norint padidinti kortelę.
- „\_n“ nuo angliško žodžio „normal“ – normalaus dydžio kortelės (86px x 86px). Standartinės kortelės naudojamos kortelių matricos pagrindiniame lange.
- „\_s“ nuo angliško žodžio „small“ – mažiausio dydžio kortelės. (50px x 50px). Jos naudojamos istorijos ir sakinio formavimo lange.

**Html** - katalogas skirtas pagrindams html šablonams saugoti. Dabar ten yra 4 šablonai „admin.html“, „login.html“, „main.html“, „new\_user.html“. kadangi sistemoje naudojama šablonų sistema. Tai išeities kodas html ir php yra skirtingose vietose. Visi html failai naudingi tik

naudojant php parašytas funkcijas ir užpildant šabloninius failus duomenimis. Kitaip šie html failai nenaudingi.

Admin.html failas skirtas atvaizduoti administruojamiems vartotojams ir jų blokavimo teisėms, bei slaptažodžių pakeitimams.

Login.html yra naudojamas pagrindiniam prisijungimo langui atvaizduoti ekrane.

Main.html yra pagrindinis programos langas prisijungus registruotam vartotojui. Čia užkraunamos kortelės meniu, puslapiavimas ir t.t.

New\_user.html skirtas iš login.html failo - vartotojui užsiregistruoti svetainėje nauju svetainės nariu. Forma su įvedimo laukais.

**Images** – tai paveikslėlių katalogas skirtas svetainės išoriniams paveikslėliams ir reikiamoms papildomoms dekoracijoms atvaizduoti. Pvz. ekrano fonui, kitiems spalvotiems paveikslėliams nefunkciniams reikalavimams įgyvendinti.

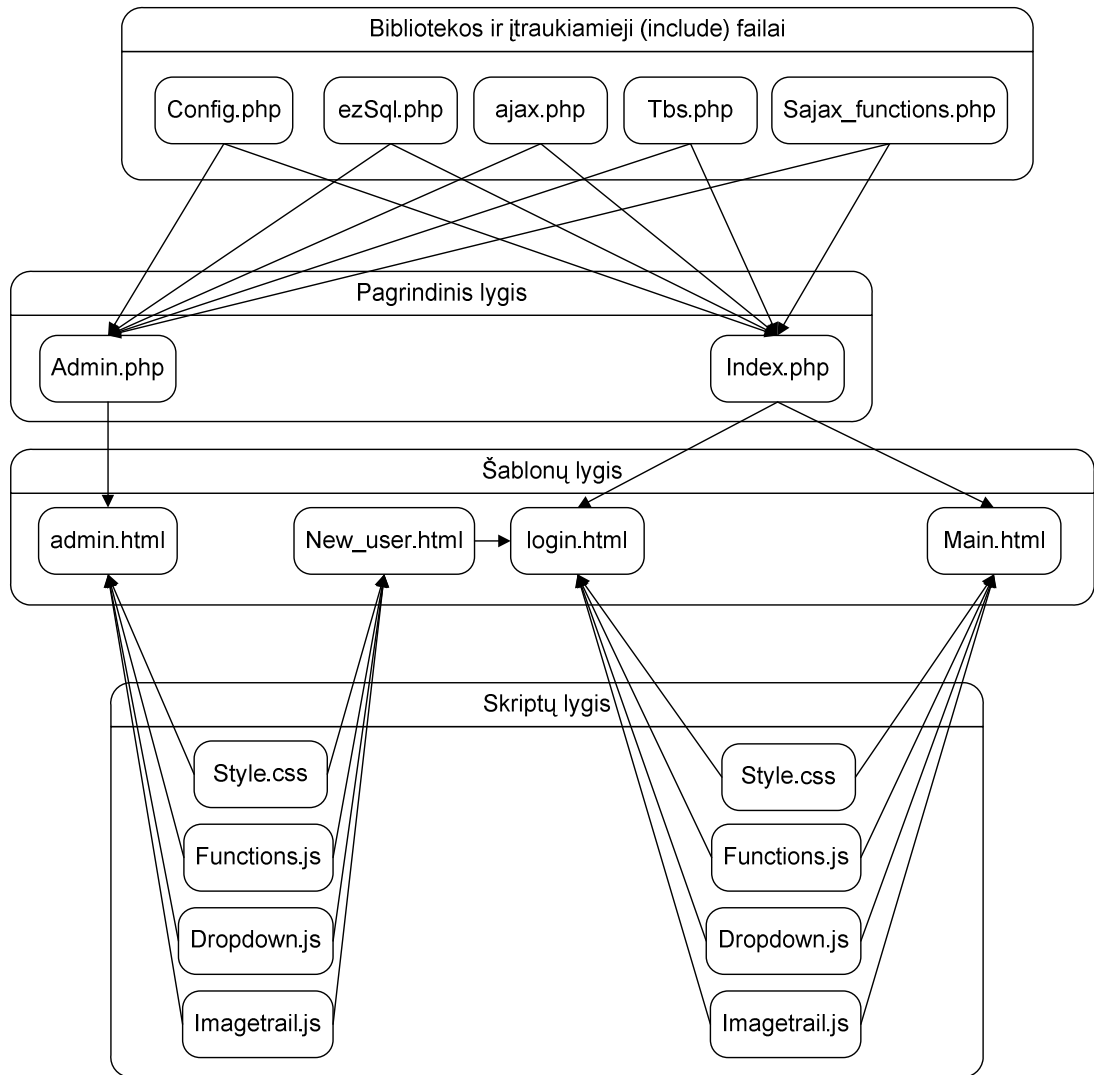
**Inc** – katalogas skirtas visoms papildomai naudojamoms klasėms įtraukti į svetainės naudojimą ir taip užtikrinti suderinamą veikimą. Direktorijos pavadinimas nusako (iš angl. įterpti, įsprausti), jog visi failai viduje bus naudojami kitų failų t.y. panaudojami pagal reikiamą paskirtį. Papildomos naudojamos nemokamos php bibliotekos:

- „ezSQL“ tai biblioteka palengvinanti darbą su MySQL duomenų baze. Galimas ir kitos duomenų bazės pritaikymas (skaityti gamintojo dokumentaciją).
- „tbs“ (tiny but strong). Šablonų panaudojimo biblioteka ji naudojama pagrindiniuose puslapiuose kur reikia HTML specialius simbolius užpildyti su php gražintus duomenis iš duomenų bazės. Detaliau galima sužinoti internete.
- Sajax simple ajax. Darbą su Ajax technologijomis palengvinanti klasė. Jos paskirtis panaudoti ajax kartu su php. Detaliau skaityti gamintojo puslapyje anglų kalba. („Simple ajax“).

Direktorijos struktūra ir klasių hierarchiškumas turėtų išlikti ir pridėdant naujų klasių ar naudojamų bibliotekų.

**Lib** – katalogas skirtas pagrindinėms ir specifinėms svetainėje naudojamoms programų dalims saugoti ir funkcionalumą įgyvendinti. Čia saugomi svarbiausi skriptai ir pridėtinės dalys tokios kaip CSS šablonų stilius, php deriniai, JavaScript skriptai.

Visi failai vienaip ar kitaip siejasi tarpusavyje. Failų struktūros sąsajų medis vaizduojamas žemiau pateikiamoje schemoje.

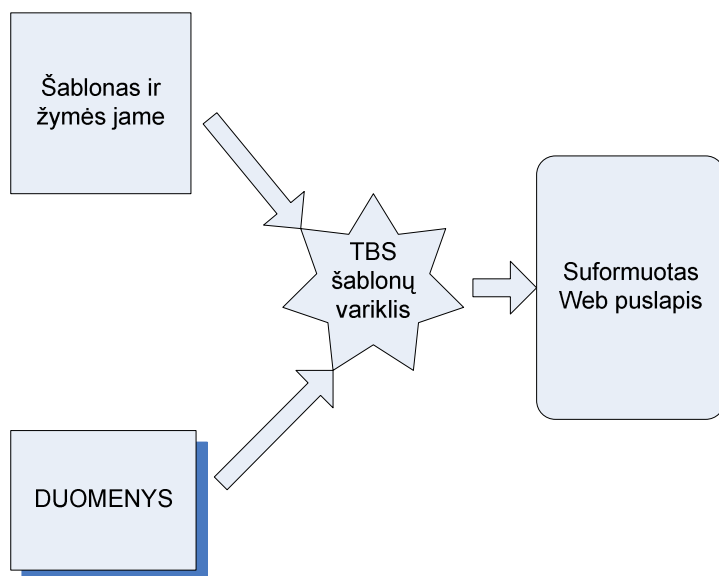


Failų hierarchija skirstoma į keletą lygių:

- *bibliotekos ir įtraukiamieji failai.* Šie failai dažniausiai naudojami visuose PHP tipo skriptų failuose. Jie įgalina Lengvųjų funkcionalumo išnaudojimą ir palengvinimą programavimo procese.
- *Pagrindinis lygis.* Tai dažniausiai paleidžiamieji failai. Juose kodo eilučių yra minimaliai tačiau tai leidžia iš failo turinio nuspręsti ką failas atlieka, kaip panaudojamos kitos bibliotekos dalys, konkrečios klasės ir kiti resursai, įgalina suprasti šablonų panaudojimo prasmę.



- *Šablonų lygis.* Tai šabloniniai failai, dažniausiai atstovaujantis „rėmus“ realiai situacijai įgyvendinti. Jie užpildomi specialiais laužtiniais skliaustais kurie savo ruožtu php pagalba pasinaudojant duomenų baze ir šabloninio valdymo klase užpildomi duomenimis. Taip gaunamas pilnavertis, funkcionalus programos langas su rezultatais. Principinė schema vaizduojama žemiau.



- *Skriptų lygis.* Tai specialios paskirties skriptai, daugiausia JavaScript. Tai skriptai reikalingi suvaldyti ir įgalinti AJAX technologiją, bei leidžianti naudoti beveik visos klaviatūros veikimą visame puslapyje.

## 5. Išities kodas ir paaiškinimai

### ezSQL bibliotekos funkcionalumas.

Pagrindiniai sistemoje naudojami ezSQL klasės metodai:

- klasės inicializavimas ir objekto sukūrimas
 

```
$db = new ezSQL_mysql();
```
- užklauso įvykdymas
 

```
$db->query("SET NAMES 'utf8'");
```
- užklauso įvykdymas ir rezultatų masyvo gražinimas

```

$sql_r = $db->get_results("SELECT car.id,
                          f_letter as raide,
                          car.f_lt pavad,
                          cat.f_lt grupe,
                          f_filename failas
                          FROM t_categories cat
                          LEFT JOIN t_cards car ON cat.id=car.f_categoryid
                          WHERE f_letter='$cardgr'
                          ORDER BY car.f_lt
                          LIMIT ".$cardpg-1*$cards_limit).",
". $cards_limit, ARRAY_A);

```

- užklauso įvykdymas ir pirmo stulpelio ir pirmos eilutės gautame rezultate gražinimas

```

$gr_kort_sk = $db->get_var("SELECT count(*) sk
                          FROM t_categories cat
                          LEFT JOIN t_cards car ON
cat.id=car.f_categoryid
                          WHERE f_letter='$cl'");

```

#### Pagrindiniai ezSQL privalumai:

- paprasta naudotis
- lengvai pritaikoma
- nemokama
- lanksti
- prisitaikoma pagal savo poreikius
- palaiko ir kitas duomenų bazes

#### Tbs bibliotekos funkcionalumas.

##### Pagrindiniai sistemoje naudojami TBS klasės metodai:

- šablono užkrovimas

```
$tbs->LoadTemplate('./html/login.html', 'utf-8');
```

- žymės priskyrimas

```
$test_zyme = "Kaip mane matote ?";
$tbs->show();
```

- bloko užpildymas

```
$tbs->MergeBlock('b11',$raides);
//kur kintamasis $raides gali buti raidžių masyvas, raidžių objektas,
//raidžių generatorius(funkcija), šaukinys į SQL duomenų bazę.
```

- šablono atvaizdavimas

```
// metodo vykdymo metu užpildomos žymes tais kintamaisiais ir tais
// blokais kurie buvo priskirti
$tpls->show();
```

Pagrindiniai TBS privalumai:

- paprasta naudotis
- lengvai pritaikoma
- nemokama
- lanksti
- mažai metodų
- funkcionali
- smarkiai pagelbsti susitvarkyti su dideliais html puslapiais

### **Sajax bibliotekos funkcionalumas.**

Pagrindiniai sistemoje naudojami sajax metodai:

- inicializuotis

```
sajax_init();
```

- užkrauti metodus php metodus

```
sajax_export("getUsers", "checkLogin");
```

- debug režimas

```
$sajax_debug=1;
```

Pagrindiniai sajax privalumai:

- paprasta naudotis
- nenaudoja klasių
- lengvai pritaikoma
- nemokama
- funkcionali
- pritaikoma
- lengva perprasti funkcionalumą

## Svarbiausių funkcijų ir failų išeities kodai su komentarais

*Index.php* – pagrindinis paleidžiamasis failas(prisijungimas, prisiregistravus permetimas į pagrindinį)

```
<?php
//sesijos pradžia (visi paleidžiamieji failai turi turėti)
session_start();
//config
$cards_limit = 40;
include_once ("./config.php");
include_once ("./inc/ezsql/ezsql.php");
include_once ("./inc/tbs/tbs.php");
include_once ("./inc/sajax/sajax.php");
include_once ("./lib/php/sajax_functions.php");

sajax_init();
//$sajax_debug_mode = 1;
sajax_export("getUsers", "checkLogin");
sajax_handle_client_request();

//$_SESSION['userid'] = 1; //defaultinis useris

$sajax_js = sajax_get_javascript();
//vartotojų prisijungimo vardai
$users_login_list = getUsers();
$raides =
array("A", "A", "B", "C", "Č", "D", "E", "E", "È", "F", "G", "H", "I", "Į", "Y", "J", "K", "L", "M", "N")

//login šablono, bloko užkrovimas
$tbs->LoadTemplate('./html/login.html', 'utf-8');
$tbs->MergeBlock('b11', $raides);
$tbs->Show();
?>
```

*Main.php* – paleidžiamas vos tik sėkmingai prisijungus per registravimosi langą. Tai pagrindinis programos langas, užkraunantis korteles ir likusią interfeiso dalį.

```
<?php
session_start();
//config
$cards_limit = 40;
include_once ("./config.php");
include_once ("./inc/ezsql/ezsql.php");
include_once ("./inc/tbs/tbs.php");
include_once ("./inc/sajax/sajax.php");
include_once ("./lib/php/sajax_functions.php");

//vartotojo prisijungimo tikrinimas
if (($SESSION['userid']!=0) && ($SESSION['userid']!= '')){
    sajax_init();
    // $sajax_debug_mode = 1;
    //sajax php funkcijų eksporto pavyzdys(kokios išorinės ajax funkcijos bus
    naudojamos šiame puslapyje)
    sajax_export("getCards", "getMenu", "addCardToChat",
"removeCardFromChatLast", "getChatLineText", "removeCardFromChat",
"getCardTabs");
    sajax_handle_client_request();
```

```

//$_SESSION['userid'] = 1; //defaultinis useris

$sajax_js = sajax_get_javascript();

//lango užpildymo informacija iš duomenų bazės seka
$user_text = getUserText();
$groups_menu = getMenu();
$cards_table = getCards();
$cards_tabs = getCardTabs($cardletter, $cardpages);
$chat_line = getChatLine();
$chat_line_text = getChatLineText();

$tbs->LoadTemplate('./html/main.html', 'utf-8');
$tbs->Show();
} else {
//neegzistuojančios sesijos atveju vartotojas metamas atgal į prisijungimo
langą.
echo "<script>window.location=\"index.php\"</script>";
//header("Location: index.php");
}
?>

```

Functions.js – javascript failas skirtas ajax saukiniam bei kitoms su puslapio funkcionalumu susijusioms dalims vykdyti.

```

/**
 * Visos parasytos funkcijos ir autorines teises priklauso
 * KTU studentui, Dariui Gudaviciui, 2006
 **/

var selRowColor = 'blue';
var selColColor = 'red';
var selCrossColor = 'yellow';

//vartotojo naršyklės atpažinimas
var strUserAgent = navigator.userAgent.toLowerCase();
var isIE = strUserAgent.indexOf("msie") > -1;
var isNS6 = strUserAgent.indexOf("netscape6") > -1;
var isNS4 = !isIE && !isNS6 && parseFloat(navigator.appVersion) < 5;

var alfa = 0;

//korteliu matricoje pazymima eilute
function selectRow(p_row){
    var reParam = /\d/;
    if (reParam.test(p_row)) {
        var eilutes =
document.getElementById(returnLetter(p_row)).getElementsByTagName('td');
    } else {
        var eilutes = document.getElementById(p_row).getElementsByTagName('td');
    }
    var i;
    for (i=0; i<eilutes.length; i++) {
        eilute = eilutes.item(i);
        if (eilute.style.backgroundColor==selRowColor){
            eilute.style.backgroundColor=selCrossColor;
            //eilute.style.padding='10px';
        }else{

```

```

        eilute.style.backgroundColor=selColColor;
    }
}
}
// korteliu matricoje pazymimas stulpelis
function selectColumn(p_col){
    var rowsai =
document.getElementById('card_table').getElementsByTagName('tr');
    if (p_col != 0){
        p_col=p_col-1;
    } else {
        p_col = 9;
    }
    for (i=0; i<rowsai.length; i++){
        vienas_rowsas = rowsai.item(i).getElementsByTagName('td').item(p_col)
        if (vianas_rowsas.style.backgroundColor==selColColor){
            vienas_rowsas.style.backgroundColor=selCrossColor;
        } else {
            vienas_rowsas.style.backgroundColor=selRowColor;
        }
    }
}

//isvaloma korteliu lentelė nuo pažymėjimu
function clearGrid(){
    var rowsai =
document.getElementById('card_table').getElementsByTagName('tr');
    for (i=0; i<rowsai.length; i++){
        columns = rowsai.item(i).getElementsByTagName('td');
        for (j=0; j<columns.length; j++){
            columns.item(j).style.backgroundColor='transparent';
        }
    }
}

//kortelės pažymėjimas vykdomas pele
function selectByMouse(td_obj){
    clearGrid();
    selectColumn(td_obj.cellIndex+1);
    selectRow(td_obj.parentNode.rowIndex+1);
}

//pagalbinė funkcija skirta vietoj paduoto skaičiaus grazinti raide(naudojama
eilutems)
function returnLetter(num){
    switch (num){
        case 1: return 'Z'; break;
        case 2: return 'X'; break;
        case 3: return 'C'; break;
        case 4: return 'V'; break;
        default: return 'Z'; break;
    }
}

/**
 * funkcija apdoroti login puslapio klaviskus
 **/
function doActionLogin(p_object, p_event){
    var val = p_object.value;
    var ilgis = val.length;
    var iKC, strKey;

```

```

if (isIE) {
    iKC = p_event.keyCode;
} else {
    iKC = p_event.which;
}

//specialus klaviatūros klavisai
var keyBackSpace = 8;
var keyEnter = 13;
var keySpace = 32;

//galimos komandos

if (ilgis == 0){           //jei nieko neparasyta
    if (iKC == keySpace){
        gebi('users_holder').onclick();
    }
    if (iKC == keyEnter){
        login('frm_uname', 'frm_password');
    }
} else if (ilgis == 1){ //jei 1 simb. parasytas
    var v1 = p_object.value.substr(0,1).toUpperCase();
    var reNum = /\d/;
    var reAZ = /[A-Z]/;

    if (reAZ.test(v1)){
        getUsersList(v1);
        p_object.value = '';
        gebi('users_holder').onclick();
    }
} else if (ilgis == 2){ //jei 2 simb. parasyti
    var v1 = p_object.value.substr(0,1).toUpperCase();
    var v2 = p_object.value.substr(1,1).toUpperCase();
    var reNum = /\d/;
    var reRaid = /[ZXCVC]/;
    var re = /[A-Z]/;

}
}

/**
 * funkcija apdoroti main puslapio klaviskus
 **/
function doAction(p_object, p_event){
    var val = p_object.value;
    var ilgis = val.length;
    var iKC, strKey;
    if (isIE) {
        iKC = p_event.keyCode;
    } else {
        iKC = p_event.which;
    }
    //gebi('cards_holder').innerHTML=iKC+'|'+val.length;

//simboliai
var keyZ = 90;
var keyX = 88;
var keyC = 67;
var keyV = 86;
var keyB = 66;
var keyN = 78;

```

```

var keyComma = 188;
var keyDot = 190;

//skaiciai
var key0 = 48;
var key1 = 49;
var key2 = 50;
var key3 = 51;
var key4 = 52;
var key5 = 53;
var key6 = 54;
var key7 = 55;
var key8 = 56;
var key9 = 57;

//specialus simboliai
var keyBackSpace = 8; //backspace
var keyEnter = 13; //enter
var keySpace = 32;

//puslapyje galimos klavisu komandos ir kombinacijos

//  backspace
//  space
//  enter
//  up
//  down

//  Z selektina eilute
//  X -----||-----
//  C -----||-----
//  V -----||-----
//  B iskviecia meniuka
//  N iskviecia mano korteles
//  1 selectina stulpeli
//  2 -----||-----
//  3 -----||-----
//  4 -----||-----
//  5 -----||-----
//  6 -----||-----
//  7 -----||-----
//  8 -----||-----
//  9 -----||-----
//  0 -----||-----
//
//  Z[0-9]
//  X[0-9]
//  C[0-9]
//  V[0-9]
//  B[A-Z]
//  1[ZXCV]
//  2[ZXCV]
//  3[ZXCV]
//  4[ZXCV]
//  5[ZXCV]
//  6[ZXCV]
//  7[ZXCV]
//  8[ZXCV]
//  9[ZXCV]
//  0[ZXCV]
if (ilgis == 0){ //jei nieko neparasyta

```



```

if (iKC == keyBackSpace){ //backspace
    removeCardFromChatLast(1);
}
if (iKC == keySpace){
    removeCardFromChatLast(17);
}
} else if (ilgis == 1){ //jei 1 simb. parasytas
    var v1 = p_object.value.substr(0,1).toUpperCase();
    var reNum = /\d/;
    //gebi('cards_holder').innerHTML=iKC+''+ilgis;
    if ((v1 == 'Z') || (v1 == 'X') || (v1 == 'C') || (v1 == 'V')){ //c
        clearGrid();
        selectRow(v1);
    } else if (reNum.test(v1)) {
        clearGrid();
        selectColumn(v1);
    } else if (v1 == 'B'){
        gebi('btn_groups').onclick();
    } else if (iKC == keySpace){
        removeCardFromChatLast(17);
        p_object.value = '';
    } else if (v1 == ','){
        switchTab(1);
        p_object.value = '';
    } else if (v1 == '.'){
        switchTab(0);
        p_object.value = '';
    } else {
        p_object.value = '';
    }
}
} else if (ilgis == 2){ //jei 2 simb. parasyti
    var v1 = p_object.value.substr(0,1).toUpperCase();
    var v2 = p_object.value.substr(1,1).toUpperCase();
    var reNum = /\d/;
    var reRaid = /[ZXC]/;
    var re = /[A-Z]/;
    //jei prima raide
    if ((v1 == 'Z') || (v1 == 'X') || (v1 == 'C') || (v1 == 'V')){
        clearGrid();
        selectRow(v1);
        if (reNum.test(v2)){
            selectColumn(v2);
            addCardToChat(getCardId_L_N(v1,v2));
            p_object.value = '';
        } else if (reRaid.test(v2) && v2 != v1){
            clearGrid();
            selectRow(v2);
            p_object.value = v2;
        }
        /*
    else if (v2==v1){
        if (alfa < 9){
            alfa = alfa+1;
        } else if (alfa == 9){
            alfa = 0;
        }
        selectColumn(alfa);*/
    } else if ((v2=='B')&&(v1!=v2)){
        gebi('btn_groups').onclick();
        p_object.value = v2;
    } else{
        p_object.value = v1;
    }
}

```

```

    }
    // jei pirma skaicius
  } else if (reNum.test(v1)){
    clearGrid();
    selectColumn(v1);
    if ((v2 == 'Z') || (v2 == 'X') || (v2 == 'C') || (v2 == 'V')){
      selectRow(v2);
      addCardToChat(getCardId_L_N(v2,v1));
      p_object.value = '';
    } else if ((v2 != v1) && (reNum.test(v2))){
      clearGrid();
      selectColumn(v2);
      p_object.value = v2;
    }
    else {
      p_object.value = v1;
    }
  } else if (v1 == 'B'){
    var re = /[A-Z]/;
    if ((re.test(v2)) && (v2!='I') && (v2!='W')){
      get_card_table(v2,1);
      gebi('btn_groups').onclick();
      p_object.value = '';
    } else {
      p_object.value = v1;
    }
  } else {
    p_object.value = '';
  }
}
}
// funkcija skirta perjungti puslapiavimo mygtukui i kita(pagrindinis ekranas)
function switchTab (p_direction){
  var tb = gebi('tab1').getAttribute('tabsu');
  var ak = gebi('tab1').getAttribute('aktyvus');
  if (p_direction == 1){ //i virsu
    if (ak*1 > 1){
      gebi('tab'+(ak*1-1)).onclick();
    }
  } else if (p_direction == 0){ //i apacia
    if (ak*1 < tb*1){
      gebi("tab"+(ak*1+1)).onclick();
    }
  }
}
}
//susirasti kortelès ID pagal simboli ir numeri(naudojama pagrindiniame
puslapyje norint išrinkti kortele kuri bus dedama i sakinio formavimo eile)
function getCardId_L_N(p_letter, p_number){
  return gebi(p_letter+'_'+p_number).getAttribute('cardid');
}
}
//palengvinimo funkcija nuo raktinio žodžio getElementById
function gebi(obj){
  return d().getElementById(obj);
}
}
//palengvinimo funkcija dažnai naudojamam objektui(pagrindiniame lange)
function ui(){
  return gebi('user_input');
}
}
//palengvinimo funkcija raktiniam document objektui
function d(){

```

```
    return document;
}
```

## Login.html – pagrindinis prisijungimo šablonas

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Vaikų su negalia abilitacijos svetainė</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<script src="lib/js/dropdown.js"></script>
<script src="lib/js/functions.js"></script>
//virtualios klaviatūros klasė
<script src="lib/js/vkeyboard.js"></script>
<style>
  body{
    margin: 0px;
    padding: 0px;
  }
  td{
    font: 6px Verdana;
  }
  .uh{
    background-color: #9999FF;
    float: left;
    margin-left: 1px;
    border: 1px solid black;
    width: 435px;
    height: 93px;
  }
  .selector{
    float: left;
    margin-left: 1px;
    cursor: pointer;
  }
  .uh_text{
    float: left;
    margin:0px;
    margin-left: 8px;
    padding:0px;
    line-height: 32px;
    height: 88px;
    font: bold 32px Verdana;
    _background-color: green;
  }
  .uh_image{
    float: left;
    margin:0px;
  }
  .uh_image img{
    border: 1px solid black;
    margin: 1px;
    padding: 0px;
  }
  .user_pass{
    float: left;
  }
  .user_pass input{
```

```

vertical-align: middle;
text-align: center;
font: 20px verdana;
line-height: 18px;
margin: 1px;
margin-top: 2px;
width: 433px;
height: 20px;
border: 1px solid black;
}
.submit{
float: left;
margin-top: 3px;
}
.usr_holder{
background-color: #FFB67F;#676767;
_float: left;
margin-left: 1px;
-border: 1px solid black;
-border-bottom: none;
border-bottom: 1px solid black;
width: 434px;
height: 93px;
cursor: pointer;
}
.usr_image{
float: left;
margin: 0px;
}
.usr_image img{
border: 1px solid black;
margin: 1px;
padding: 0px;
}
.usr_text{
float: left;
margin: 0px;
margin-left: 8px;
padding: 0px;
line-height: 32px;
height: 88px;
font: bold 32px Verdana;
}
.usr_text div{
margin-top: 18px;
font: bold 26px Verdana;
color: #007FA3;
}
#users_dropdown{
position: absolute;
visibility: hidden;
width: 436px;
border: 1px solid black;
height: 286px;
overflow: auto;
background-color: white;
padding: 0px;
}
.abc{
font: 16px Verdana;
cursor: pointer;
}

```

```

}
.abc a{
  text-decoration: none;
  color: red;
  padding-left: 4px;
  padding-right: 4px;
  display: marker;
}
.abc a:hover{
  background-color: #CCCC99;
  border: 1px solid black;
  padding-left: 3px;
  padding-right: 3px;
}
</style>
<script>
  //pagrindinės ajax funkcijos
  [var.sajax_js;htmlconv=no;protect=no]
  //kreipimosi i ajax procedura pavyzdys(kai ja kas nors is vartotojo sasajos
iskviecia)
  function getUsersList(p_letter){
    x_getUsers(p_letter, return_users_list);
  }
  //ajax grazinto vartotoju sarašo sutvarkymo ir padėjimo su javascript pavyzdys
  function return_users_list(r){
    gebi('users_dropdown').innerHTML = r;
  }
  //perdirbto vartotojo ivedimo „combo box“ apiforminimas
  function setUserName(p_obj, p_uname){
    var src_img = p_obj.getElementsByTagName('img');
    var dest_img = gebi('uh').getElementsByTagName('img');
    dest_img[0].src = src_img[0].src;
    gebi('uh').style.backgroundColor='white';
    gebi('frm_uname').value = p_uname;
    gebi('uname_text').innerHTML = p_uname;
  }
  //iskrentancio meniu nustatymas i pradine busena
  function resetUserHolder(){
    gebi('uname_text').innerHTML = "Pasirinkite vardą";
    var dest_img = gebi('uh').getElementsByTagName('img');
    dest_img[0].src = 'images/no_user_pic.gif';
    gebi('uh').style.backgroundColor='#4B71D4';
  }
  //ajax logino kvietimas
  function login(p_uname, p_password){
    var un = gebi(p_uname).value;
    var pw = gebi(p_password).value;
    //ajax funkcijos issaukimas su parametrais is vartotojo sasajos
    x_checkLogin(un,pw,return_user_status);
  }
  //pranesimo apie klaidas formavimas
  function return_user_status(r){
    var str = r.split("||");
    var login = str[0];
    var paaiskinimas = str[1];
    if (login*1 == 1){
      window.location="./main.php";
    } else {
      gebi('frm_password').value='';
      gebi('user_status').innerHTML=paaiskinimas;
      gebi('user_status').style.backgroundColor='#FF947F';
    }
  }

```

```

        setTimeout('gebi(\'user_status\').innerHTML=\'\';
gebi(\'user_status\').style.backgroundColor=\'transparent\'',3000);
    }
}
</script>
//html prisijungimo formos kodas
<body background="images/bg5.jpg" onLoad="ui().focus();" onClick="ui().focus();"
<div id="users_dropdown">[var.users_login_list;htmlconv=no;protect=no]</div>
<table id="main" align="center" width="1003" height="687" border="0"
style="border: 1px solid black;" cellpadding="0" cellspacing="0">
    <tr>
        <td colspan="7" width="1003" height="63"><input id="user_input"
type="text" style="height: 6px; width: 10px; background-color: transparent;
border:none; font: 4px verdana;" maxlength="1" onKeyUp="doActionLogin(this,
event); this.focus();">&nbsp;</td>
        <td height="63" nowrap></td>
    </tr>
    <tr>
        <td rowspan="6" width="86" height="624">&nbsp;</td>
        <td colspan="3" width="800" height="136">
            </td>
        <td colspan="3" rowspan="4" width="117" height="551">&nbsp;</td>
        <td height="136" nowrap></td>
    </tr>
    <tr>
        <td colspan="3" width="800" height="46" align="center">&nbsp;</td>
        //bloko b11 pavyzdys „[]“ skliaustuose
        <span class="abc" title="Klavišas [b11.val;block=span]"
onClick="getUsersList('[b11.val]'); gebi('uh').onclick();"><a href=""
onClick="return false;">[b11.val]</a>&nbsp;</span>
        </td>
        <td height="46" nowrap></td>
    </tr>
    <tr>
        <td rowspan="4" width="182" height="442">&nbsp;</td>
        <td width="454" height="122">
            <div id="uh" class="uh">
                <div class="uh_image"></div>
                <div id="uname_text" class="uh_text">Pasirinkite vardą</div>
            </div>
            <div id="selector" class="selector" onClick="gebi('uh').onclick();"></div>
            <div class="user_pass"><input style="z-index:1" type="password"
name="frm_password" value="" onKeyDown=""></div>
            <div class="submit" onClick="login('frm_uname', 'frm_password');"></div>
            <input id="frm_uname" type="hidden">
        </td>
        <td rowspan="4" width="164" height="442" valign="top"><div style="margin-
top: 96px;">&nbsp;</div><a href="registrer?" style="margin-left: 30px; font:
12px verdana;" title="Registruotis į portalą">Naujas vartotojas?</a></td>
        <td height="122" nowrap></td>
    </tr>
    <tr>
        <td rowspan="3" width="454" height="320" align="center" valign="top">
            <div id="user_status" style="font: bold 14px verdana; color: #171EB2;
line-height: 36px; letter-spacing: 1px;">&nbsp;</div></td>
        <td height="247" nowrap></td>
    </tr>
    <tr>
    </tr>

```



```

    x_removeCardFromChat(p_cardid, return_chat_line);
    uf();
}
function return_chat_line(r){
    document.getElementById('chat_holder').innerHTML = r;
    x_getChatLineText(return_chat_line_text);
}
function return_chat_line_text(r){
    document.getElementById('chat_text_holder').innerHTML = r;
}
//userio ivedimo fokusavimas
function uf(){
    ui().focus();
}
//naudojama uzpildyti karts nuo karto atsirandančias laukimo spragas()del
interneto trigdziu
function sajax_loading(){
    document.getElementById('cards_holder').innerHTML="<img
src='./images/loading.gif' alt=''>";
}
</script>
<body onload="ui().focus();" onclick="ui().focus();"
onMouseMove="ui().focus();">
//paslėptas nuo vartotojo ivedimo laukas kuris turi buti pastoviai sufokusuotas
<input type="input" id="user_input" maxlength="2" class="user_input"
onKeyPress="this.onkeydown();" onKeyUp="doAction(this, event); this.focus();">
[var.groups_menu/htmlconv=no;protect=no]
<span class="user_text">[var.user_text/htmlconv=no;protect=no]</span>
<table id="slice_master_table" width="1003" height="687" border="0"
cellpadding="0" cellspacing="0" align="center">
<tr>
//menu vieta
<td id="menu_holder" width="73" height="429" valign="top"
class="slice_menu_container">
<div id="btn_groups" class="btn_groups">
<span style="font: bold 11px verdana;
color:#FF614C;">B</span>&nbsp;  GRUPĖS
</div>
<div class="btn_my">
<span style="font: bold 11px verdana;
color:#FF614C;">N</span>&nbsp;  MANO
</div>
</td>
<td width="930" height="429" class="slice_main_container" valign="top">
<table align="left" cellpadding="0" cellspacing="0" border=0
id="container_table">
//korteliu matricos užkrovimo vieta(kai uzkraunama per ajax)
<tr class="container_num_row">
<td colspan="class="container_corner">&nbsp;  </td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td style="width: 13px; background-color: transparent; border-
bottom: none;">&nbsp;  </td>

```





jos įvykdomos automatiškai) per Sajax klasę.

```
<?php
/**
 * grazina korteliu matrica lentele su paveiksliukais
 **/
function getCards($cardgr = "T", $cardpg = 1) {
    global $db, $tbs, $cardletter, $cardpages, $cards_limit;
    $cardletter = $cardgr;
    $cardpages = $cardpg;
    $sql_r = $db->get_results("SELECT car.id,
                                f_letter as raide,
                                car.f_lt pavad,
                                cat.f_lt grupe,
                                f_filename failas
                                FROM t_categories cat
                                LEFT JOIN t_cards car ON cat.id=car.f_categoryid
                                WHERE f_letter='$cardgr'
                                ORDER BY car.f_lt
                                LIMIT ".$cardpg." ".$cards_limit,
    ARRAY_A);
    $eiles = array('Z', 'X', 'C', 'V');
    $irasu = 10;
    $rez = '<table align="left" cellpadding="0" cellspacing="0" border=0
id="card_table">'. "\n";
    $nr = 1;
    $j = 1;
    foreach ($eiles as $k=>$v) {
        $rez .= '<tr id="'. $v. "'>'. "\n";
        for ($i=1; $i <= $irasu; $i++){
            $vdb = $sql_r[$j-1];
            if ($i == 10){
                $i = 0;
            }
            if ($vdb['id'] != ''){
                $tmp = ' <td id="{raide_tarpas_sk}"
                    cardid="{cardid}"
                    onClick = "addCardToChat({cardid});"
                    onMouseOver = "selectByMouse(this); timer{raide_sk} =
setTimeout(\'showtrail(\\\'\\\'./cards/{raide}_b/{failas}.gif\\\'\\\'', \\\'\\\'{pavad}\\\',
\\\'\\\'{grupe}\\\', 1)\\\'\\',500);"
                    onMouseOut = "hidetrail();"
clearTimeout(timer{raide_sk});">
                    <div class = "card_container"><div
class="card_caption">{pavad}</div></div>
                    </td>'. "\n";
                $final = str_replace("{raide_tarpas_sk}", $v.'_'. $i, $tmp);
                $final = str_replace("{cardid}", $vdb['id'], $final);
                $final = str_replace("{raide_sk}", $vdb['raide']. $j , $final);
                $final = str_replace("{raide}", $vdb['raide'], $final);
                $final = str_replace("{failas}", $vdb['failas'], $final);
                $final = str_replace("{pavad}", $vdb['pavad'], $final);
                $final = str_replace("{grupe}", $vdb['grupe'], $final);
                $rez .= $final;
            } else {
                $tmp = ' <td id="{raide_tarpas_sk}"
                    cardid="{cardid}"
                    onMouseOver="clearGrid();"
                    onMouseOut="">
                    <div class="card_container_none">&nbsp;<div
```

```

class="card_caption-">&nbsp;</div></div>
        </td>'. "\n";
        $final = str_replace("{raide_tarpas_sk}", $v.'_'. $i, $tmp);
        $final = str_replace("{cardid}", '-1', $final);
        $rez .= $final;
    }
    $j++;
    if ($i == 0){
        $i = 10;
    }
}
$rez .= "\n".'grazina korteliu puslapius apskaiciutus pagal kiekvienos grupes poreikius
 **/
function getCardTabs($cl='T', $cp=1){
    global $db, $cards_limit;
    //ezSQL panaudojimo pavyzdys
    $gr_kort_sk = $db->get_var("SELECT count(*) sk
                                FROM t_categories cat
                                LEFT JOIN t_cards car ON cat.id=car.f_categoryid
                                WHERE f_letter='$cl'");
    $ciklu = ceil($gr_kort_sk/$cards_limit);
    //echo $gr_kort_sk;
    if ($cp > $ciklu){
        $cp = $ciklu;
    }
    $colors = array('#FFC800', 'darkmagenta', 'greenyellow', 'gray');
    $i=1;
    while ($i <= $ciklu){
        if ($i == $cp){
            $content = "<span style='color: red; font-size: 10px; font-weight:
bold;'>&lt;&br>&gt;</span><br><br><br>&nbsp;". $i;
            $style = "background-color: black; color: white;";
            $aktyvus = $i;
        } else {
            $content = "&nbsp;<br><br><br><br>&nbsp;". $i;
            $style = 'background-color: '.$colors[$i-1].';';
        }
        $str .= '<div id="tab'. $i.'" class="card_tab" style="'. $style.'"
tabsu="'. $ciklu.'" aktyvus="{aktyvus}" onClick="get_card_table('\''.$cl.'\'',
'. $i.')">'. $content.'grazina div elemente lentele su meniu grupemis ir klaviaturos saukiniais
 **/
function getMenu(){
    global $db;
    $tmp = $db->get_results("SELECT * FROM t_categories ORDER BY f_letter",
ARRAY_A);
    $items_count = sizeof($tmp);
    $left_items = ceil($items_count/2);
    // $right_items = $items_count - $left_items;
    $rez = "<div id='groups_menu' style='position: absolute; visibility:

```

```

hidden;'>";
$rez .= '<table border="0" cellpadding="0" cellspacing="0">';
$r = $left_items;
for ($l = 0; $l < $left_items; $l++){
    $v = $tmp[$l];
    $rez .= "<tr>";
    $rez .= "<td><a href='#' onclick='get_card_table(\"\".$v['f_letter'].\"\", 1);
return false;'><span>\".$v['f_letter'].\"</span>&nbsp;\".$v['f_lt'].\"</a></td>";
    if ($r < $items_count){
        $v = $tmp[$r++];
        $rez .= "<td><a href='#' onclick='get_card_table(\"\".$v['f_letter'].\"\",
1); return
false;'><span>\".$v['f_letter'].\"</span>&nbsp;\".$v['f_lt'].\"</a></td>";
    } else {
        $rez .= "<td>&nbsp;</td>";
    }
    $rez .= "</tr>";
}
$rez .= "</table>";
$rez .= "</div>";
return $rez;
}
/**
 * grazinama pokalbio sakinio formavimo linija su ten esančiomis kortelėmis
 **/
function getChatLine(){
    global $db;
    $sql = "SELECT car.id,
                                f_letter raide,
                                f_filename failas,
                                car.f_lt pavad,
                                cat.f_lt grupe
                                FROM t_line lin
                                INNER JOIN t_cards car ON car.id=lin.f_cardid
                                INNER JOIN t_categories cat ON
cat.id=car.f_categoryid
                                WHERE lin.f_userid=\".$_SESSION['userid'].
                                \" ORDER BY lin.id";

    $tmp = $db->get_results($sql, ARRAY_A);
    if (sizeof($tmp) > 0){
        $i = 0;
        // $final = '<div style="width:300px; border: 2px solid blue;
float:left;">&nbsp;</div>';
        $final = ''; $rez = '';
        foreach ($tmp as $key=>$vdb) {
            $final = '';
            $final = str_replace("{id}", $vdb['id'], $final);
            $final = str_replace("{raide_sk}", $vdb['raide']."_chat_". $i++ ,
$final);

            $final = str_replace("{raide}", $vdb['raide'], $final);
            $final = str_replace("{failas}", $vdb['failas'], $final);
            $final = str_replace("{pavad}", $vdb['pavad'], $final);
            $final = str_replace("{grupe}", $vdb['grupe'], $final);

```

```

        $rez .= "\n".$final;
    }
    return $rez."\n";
}
return "&nbsp;";
}
/**
 * grazinamas pokalbio žodžio formavimo tekstas
 **/
function getChatLineText(){
    global $db;
    $sql = "SELECT car.id,
            f_letter raide,
            f_filename failas,
            car.f_lt pavad,
            cat.f_lt grupe
            FROM t_line lin
            INNER JOIN t_cards car ON car.id=lin.f_cardid
            INNER JOIN t_categories cat ON cat.id=car.f_categoryid
            WHERE lin.f_userid=".$_SESSION['userid']." ORDER BY lin.id";
    $tmp = $db->get_results($sql, ARRAY_A);
    if (sizeof($tmp) > 0){
        $i = 0;
        $final = ''; $rez='';
        foreach ($tmp as $key=>$vdb) {
            $final = '{pavad}';
            $final = str_replace("{pavad}", $vdb['pavad'], $final);
            $rez .= "\n".$final;
        }
        return "<span class='line_text'>".$rez."</span>\n";
    }
    return "&nbsp;";
}
}
/**
 * idedama kortele i pokalbio sakinio formavimo eile
 **/
function addCardToChat($p_cardid = ''){
    global $db;
    if ($p_cardid != ''){ //jei netuscia
        $skiek_linijoje_korteliu = $db->get_var("SELECT count(*) FROM t_line WHERE
f_userid=".$_SESSION['userid']);
        $last_card = $db->get_var("SELECT f_cardid FROM t_line WHERE
f_userid=".$_SESSION['userid']." ORDER BY id DESC LIMIT 1");
        //echo $last_card;
        if ($p_cardid != $last_card) { //jei nesikartoja
            if ($skiek_linijoje_korteliu < 17){ //jei nevirsija leistinu korteliu
skaicius
                $db->query("INSERT INTO t_line (f_cardid, f_userid) VALUES
(".$p_cardid.", ".$_SESSION['userid'].")");
            }
        }
        return getChatLine();
    }
}
}
/**
 * ismetama paskutine(s) kortele(s) is chat linijos (backspace)
 * @p_num jei nieko nepaduodama istrinama paskutine nuo galo
 **/
function removeCardFromChatLast($p_num = 1){
    global $db;
    if ($p_num != '') {

```

```

    for($i = 1; $i <= $p_num; $i++){
        $lineid = $db->get_var("SELECT id FROM t_line WHERE
f_userid=".$_SESSION['userid']." ORDER BY id DESC LIMIT 1");
        //echo $lineid;
        if ($lineid != ''){
            $db->query("DELETE FROM t_line WHERE id=".$lineid);
        }
    }
    return getChatLine();
}
}
}
/**
 * ismetama kortele is chat linijos (pasirinktoji) - kai vykdomas peles
 * paspaudimas
 **/
function removeCardFromChat($p_cardid = 0){
    global $db;
    $db->query("DELETE FROM t_line WHERE f_userid=".$_SESSION['userid']." and
f_cardid = ".$p_cardid);
    return getChatLine();
}
/**
 * sugrazinamas vartotojo prisijungimo vardas
 **/
function getUserText(){
    global $db;
    return $db->get_var("SELECT CONCAT('(' ,f_uname,') ',f_name,' ',f_surname) FROM
t_users WHERE id=".$_SESSION['userid']);
}
/**
 * grazina vartotoju vardus(pagrindiniame login lange) iskrentanciamie meniu
 **/
function getUsers($p_letter='A'){
    global $db;
    $u_list = $db->get_results("SELECT * FROM t_users WHERE f_disabled != 1 and
f_type=0 and f_name LIKE '". $p_letter."%' ", ARRAY_A);
    if (sizeof($u_list) > 0){
        //$tmp = '<div id="users_dropdown">';
        foreach ($u_list as $k=>$v) {
            if ($v['f_file'] != ""){
                $failas = "user_images/" . $v['f_file'];
            } else {
                $failas = "images/no_user_pic.gif";
            }
            $str = '<div class="usr_holder" onClick="setUserName(this, \''{uname}\')';
gebi('\users_dropdown\').onmouseout();">
                <div class="usr_image"></div>
                <div class="usr_text">{uname} <br><div>{name}
{surname}</div></div>
                </div>';
            $str = str_replace("{uname}", $v['f_uname'], $str);
            $str = str_replace("{failas}", $failas, $str);
            $str = str_replace("{name}", $v['f_name'], $str);
            $str = str_replace("{surname}", $v['f_surname'], $str);
            $tmp .= $str;
        }
        //$tmp .= '</div>';
    }
    return $tmp."<br>";
}
/**

```

```

* autentifikuoja ir autorizuoja vartotoja prie sistemas
**/
function checkLogin($p_user, $p_pass){
    global $db;
    // $script = "<script></script>";
    if (($p_user != "" ) and ($p_pass != "")){
        $useris = $db->get_row("SELECT * FROM t_users WHERE f_urname='".$p_user.'"
and f_pass='".$md5($p_pass)."', ARRAY_A);
        if ($useris['id'] != ''){
            if ($useris['f_blocked']!=1){
                $_SESSION['userid'] = $useris['id'];
                //todosurasyti info apie useri i loga
                return "1|| ";
            }
            else{
                return "0||".$script."Vartotojas blokuotas. Susisiekite su svetainės
administratoriumi.";
            }
        } else{
            return "0||".$script."Neteisingas prisijungimo vardas ir/arba slaptažodis
!";
        }
    } else {
        return "0||".$script."Patikrinkite prisijungimo duomenis.";
    }
}
?>

```