

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Rolandas Karosas

**Uždavinių generavimas ir jų sprendimo automatinis
tikrinimas**

Magistro darbas

Darbo vadovas

doc. dr. E. Karčiauskas

Kaunas, 2004

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

TVIRTINU

Katedros vedėjas

dr. E. Bareiša

2004 05

**Uždavinių generavimas ir jų sprendimo automatinis
tikrinimas**

Informatikos mokslo magistro baigiamasis darbas

Kalbos konsultantė

Lietuvių kalbos katedros lektorė

dr. J. Mikelionienė

2004 05

Vadovas

doc. dr. E. Karčiauskas

2004 05

Recenzentas

doc. K. Baniulis

2004 05

Atliko

IFM-8/2 gr. stud.

R. Karosas

2004 05

Kaunas, 2004

Turinys

| | | |
|--------|--|----|
| 1. | Įvadas | 6 |
| 2. | Analitinė dalis | 7 |
| 2.1 | Tikslas | 7 |
| 2.2 | Problemos | 7 |
| 2.3 | Galimi sprendimai | 7 |
| 2.3.1 | Panašių sistemų apžvalga | 8 |
| 2.4 | Priimto sprendimo pagrindimas | 14 |
| 2.4.1 | Priimto sprendimo tikslai | 14 |
| 2.4.2 | Uždavinių generavimas | 14 |
| 2.4.3 | Rezultatų automatinis tikrinimas | 15 |
| 2.4.4 | Užduočių interaktyvumas | 15 |
| 2.5 | Projektavimo metodologijos ir technologijų analizė | 15 |
| 2.5.1 | Servletai | 15 |
| 2.5.2 | Java ir Java Beans | 16 |
| 2.5.3 | DBVS | 16 |
| 2.5.4 | CASE įrankiai | 17 |
| 3. | Projektinė dalis | 18 |
| 3.1 | Tikslas | 18 |
| 3.2 | Veiklos sfera | 18 |
| 3.3 | Bendra apžvalga | 18 |
| 3.4 | Architektūrinis vaizdas | 19 |
| 3.5 | Architektūriniai tikslai ir apribojimai | 19 |
| 3.6 | Panaudojimo atvejų vaizdas | 19 |
| 3.6.1 | Panaudojimo atvejų realizacijos | 19 |
| 3.7 | Loginis vaizdas | 25 |
| 3.7.1 | Apžvalga | 25 |
| 3.7.2 | Vartotojo klasės | 26 |
| 3.7.3 | Administratoriaus klasės | 27 |
| 3.7.4 | Objektų klasės | 28 |
| 3.7.5 | DB ir servletų klasės | 29 |
| 3.8 | Proceso vaizdas | 30 |
| 3.8.1 | Uždavinių generavimo ir jų sprendimo tikrinimo sistemos procesas | 30 |
| 3.8.2 | Vartotojo kūrimo sekos diagrama | 31 |
| 3.8.3 | Vartotojo redagavimo sekos diagrama | 31 |
| 3.8.4 | Uždavinio atlikimo sekos diagrama | 32 |
| 3.9 | Paskirstymo vaizdas | 33 |
| 3.10 | Realizacijos vaizdas | 33 |
| 3.10.1 | Apžvalga | 33 |
| 3.10.2 | Komponentų sąveikos diagrama | 34 |
| 3.10.3 | Duomenų vaizdas | 35 |
| 3.11 | Dydis ir našumas | 35 |
| 3.12 | Apibendrinimas | 35 |

| | | |
|--------|---|----|
| 4. | Tyrimo dalis | 36 |
| 4.1 | Paskirtis | 36 |
| 4.2 | Realiai atlikto darbo kokybės analizės tikslai | 36 |
| 4.3 | Funkcionalumo tyrimas | 36 |
| 4.4 | Standartų laikymasis | 37 |
| 4.5 | Sistemos našumas | 37 |
| 4.6 | Sistemos patikimumas | 38 |
| 4.7 | Sistemos testuojamumas | 38 |
| 4.8 | Vartotojo sąsaja | 38 |
| 4.9 | Sistemos saugumas | 38 |
| 4.10 | Kokybės vertinimo procesas | 38 |
| 4.10.1 | Interviu su užsakovu | 38 |
| 4.10.2 | Apklausų anketos | 39 |
| 4.11 | Vertinimo rezultatai | 39 |
| 5. | Eksperimentinė dalis | 40 |
| 5.1 | Eksperimento tikslas ir uždaviniai | 40 |
| 5.2 | Eksperimento objektas | 40 |
| 5.3 | Eksperimentinio tyrimo eiga | 40 |
| 5.3.1 | Vartotojų požiūris į kompiuterines žinių tikrinimo sistemas | 40 |
| 5.3.2 | Žinių tikrinimo sistemų savybių tyrimas | 41 |
| 5.4 | Eksperimento apibendrinimas | 41 |
| 6. | Išvados | 43 |
| 7. | Literatūra | 44 |
| 8. | Terminų ir santrumpų žodynas | 45 |
| 9. | Priedai | 46 |

SUMMARY

The tasks generating and their answers automatic checking problems of web-based assessment and testing systems was formulated in this document. The main tasks for thesis were set. The similar systems were analyzed and possible solutions reviewed. In this case, was decided to design the tasks generating and automatic checking software, which will resolve the tasks, brought to user's recurrence, improve the skills checking. The technologies for this project realization were choused.

The created software system was tested and the quality analysis was made. Finally, the experimental research was accomplished. All the research results showed that the raised problems were resolved.

Santrauka

Magistro darbo metu buvo suformuluotos žinių bei įgūdžių tikrinimo sistemų pateikiamų uždavinių generavimo ir jų sprendimų automatinio tikrinimo problemos, iškelti pagrindiniai darbo tikslai. Atlikus panašių programų sistemų, bei galimų sprendimų analizę buvo nuspręsta realizuoti uždavinių generavimo ir jų sprendimo tikrinimo programų sistemą, kuri išspręstų užduočių pateikiamų vartotojams kartojimąsi, pagerintų bei užtikrintų efektingesnį įgūdžių tikrinimą. Remiantis priimtais sprendimais buvo suprojektuota programų sistema, pasirinktos priemonės bei technologijos projekto įgyvendinimui.

Realizuota programų sistema buvo ištestuota, atlikta jos kokybės analizė, bei tyrimai. Gauti rezultatai parodė, kad darbo metu iškelti tikslai buvo įgyvendinti ir realizuota programų sistema išsprėdė uždavinių generavimo ir jų sprendimo automatinio tikrinimo problemas.

1. Įvadas

Dabartiniu metu sparčiai plintant interneto tinklo paslaugoms, mokyme pereinama į kitą, kokybiškai aukštesnį lygį. Išsivystė nuotolinio mokymo sistemos. Tokiose sistemose informacija skleidžiama klasikiniu vienas su daug (angl. *one-to-many*) principu, tačiau jose sukuriama virtualią mokymosi erdvę, vartotojai tiesiogiai nedalyvaujantys paskaitose gali dalyvauti bendrame mokymo procese jiems tinkamu metu ir jiems tinkamoje vietoje. Tam pakanka prisijungti prie nuotolinio mokymo sistemos, naudojant internetinę naršyklę. Įgytas žinias visada reikia įvertinti. Žinių vertinimas gali būti integruotas į mokymo sistemas, tačiau egzistuoja ir atskiros nuotolinės žinių tikrinimo sistemos, kuriose akcentuojamas įgytų žinių ir/arba praktinių įgūdžių egzaminavimas ir testavimas, o ne mokymas. Vertinant įgytas žinias ar įgūdžius nuotoliniu būdu dažniausiai remiamasi testų sudarymo principu, kuriuose yra pateikiamos užduotys, o atsakymai į jas byloja apie egzaminuojamojo žinių lygį. Pateikiamus uždavinius galima suskirstyti į keletą tipų: klausimai su keletą atsakymų variantų, klausimai su atsakymais iš žemyn iškrentančio meniu (angl. *drop down*), klausimai, kur atsakymą reikia įrašyti, klausimai, kuriuose vienas sąrašas turi atitikti kitą, klausimai, kuriuose atsakymą reikia „nutempti“ į teisingą vietą (angl. *drag and drop*), klausimai kuriuose kaip teisingą atsakymą reikia pažymėti „karštą vietą“ (angl. *hot spot*). Greitai populiarėjant nuotoliniam mokymui ir augant nuotolinių sistemų vartotojų skaičiui, priverstos vystytis ir žinių vertinimo sistemos – siekiama optimizuoti egzaminavimo procesą, užtikrinti savalaikį įvertinimą, kokybišką ir objektyvų žinių ir įgūdžių patikrinimą. Ieškoma būdų, kaip išvengti pastovaus tų pačių užduočių pateikimo, kaip realizuoti praktinių įgūdžių patikrinimą, kad jis būtų kuo efektyvesnis ir reikalautų iš vartotojo pademonstruoti kuo daugiau savo įgytų praktinių žinių.

Toliau šiame dokumente pateikiami darbo uždaviniai, iškilusios inžinerinės problemos, jų sprendimo būdų analizė, sprendimų apžvalga, realizuoto sprendimo pagrindimas bei įvertinimas.

Atliekant analizę pagrindinis dėmesys skirtas kompiuterinio išprusimo žinių vertinimo sistemoms.

2. Analitinė dalis

2.1 Tikslas

Pagrindinis šios darbo dalies tikslas – išanalizuoti žinių vertinimo sistemų uždavinių pateikimo, generavimo būdus, metodus, optimizavimo galimybes, išryškinti uždavinių sprendimų tikrinimo, pateikimo, generavimo esmines problemas, apžvelgti bei rasti sprendimus iškeltoms problemoms spręsti.

2.2 Problemos

Testuojant vartotojų kompiuterinį išprusimą yra tikrinamas gebėjimas naudotis kompiuteriu bei programine įranga. Kompiuterio bei programinės įrangos naudojimo teorinės žinios gali būti tikrinamos pateikiant klausimus ar paveikslėlius ir galimus atsakymų variantus, paveikslėlius, kuriuose reikia pažymėti teisingą atsakymą. Testuojant praktinius vartotojų įgūdžius taip pat gali būti pateikiami klausimai ar paveikslėliai ir galimi atsakymų variantai, paveikslėliai, kuriuose reikia pažymėti teisingą atsakymą arba užkraunama tikrinamos programinės įrangos darbo aplinka, kurioje reikia atlikti realius veiksmus užduočiai atlikti.

Kompiuterinio raštingumo žinių testavimui pateikiamas nustatytas uždavinių skaičius. Sistemų administratoriams iškyla problema kaip išvengti atsakymų „nutekėjimo“ ar „nusirašinėjimo“, tų pačių klausimų pastovaus kartojimosi, nepaisant to, jei ir klausimų pateikimo eiliškumas yra automatiškai generuojamas. Tenka keisti pateikiamus klausimus.

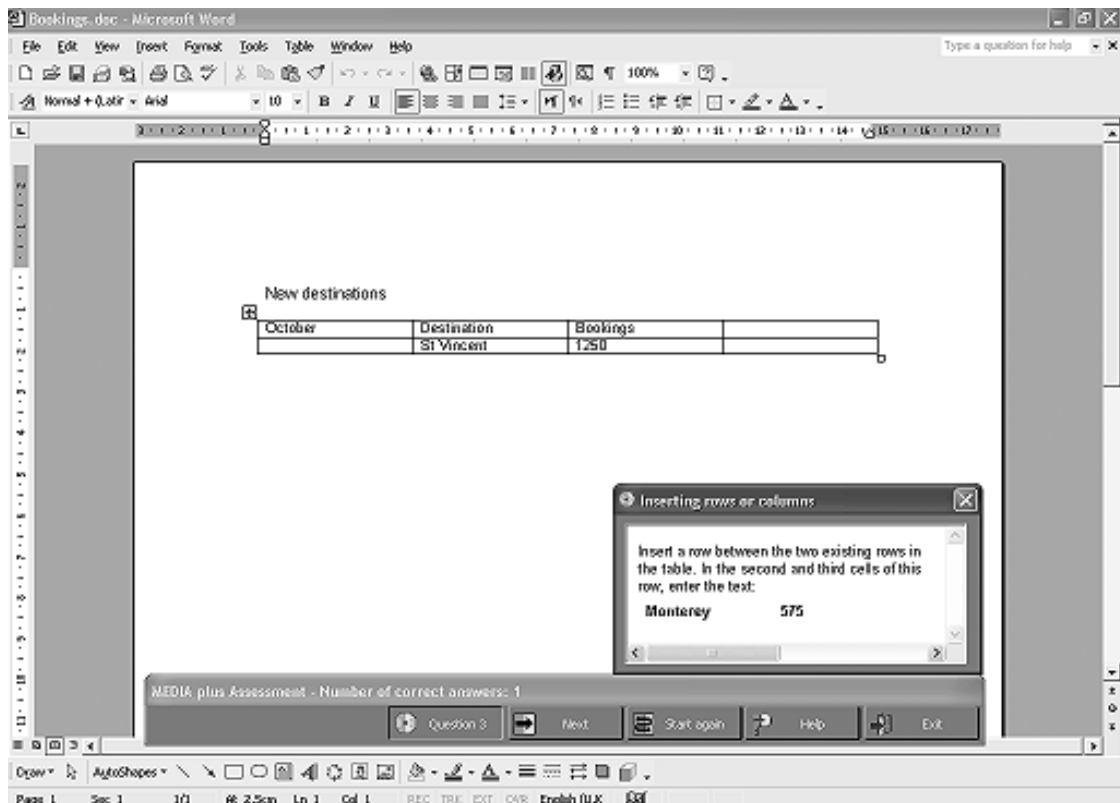
Dar viena problema – kaip vartotojui pateikti uždavinį, kad maksimaliai būtų patikrintos jo žinios, t.y. ar pateikti kokios nors programinės įrangos vaizdo iškarpą ir pateikti galimus atsakymų variantus, ar pateikti vaizdo iškarpą reikalaujant vartotojo pažymėti joje kažkokį tai aktyvų lauką, ar pateikti realios programinės įrangos darbo lauką ir suteikti vartotojui galimybę atlikti veiksmus analogiškus galimiems PĮ veiksmams. Kaip kiekvienu konkrečiu atveju realizuoti uždavinių generavimą, bei įgalinti automatinį sprendimų tikrinimą? Kokias programavimo technologijas pasirinkti realizuojant sprendimus?

2.3 Galimi sprendimai

Šioje dalyje apžvelgiamos panašios sistemos Lietuvoje ir užsienyje, nagrinėjami galimi ir priimti problemų sprendimai.

2.3.1 Panašių sistemų apžvalga

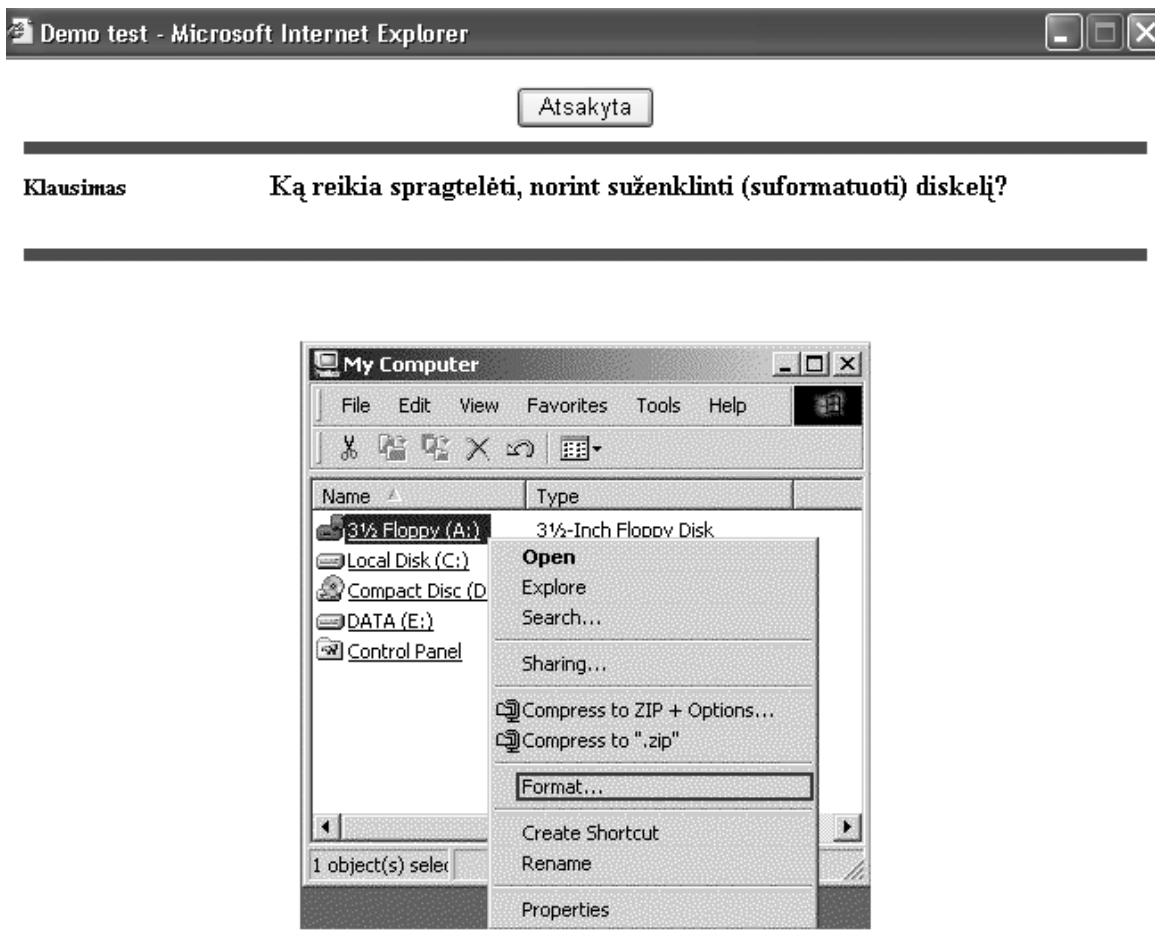
„MEDIPlus“ programų sistema, kuri skirta „Microsoft Office“ įvairių versijų paketo programų įgytų žinių tikrinimui [9]. Nors ši sistema sukurta konkretaus produkto įgūdžių testavimui, tačiau ją galima įvertinti aukščiausiu balu. „MEDIPlus“ įrankis veikia Internet tinkle ir nereikalauja jokio papildomo įdiegimo, išskyrus tai, kad kliento kompiuteryje turi būti įdiegta „Microsoft Office“ atitinkamos versijos programinė įranga. Uždavinių atlikimas yra pilnai interaktyvus. Paleidus įrankį sistema kliento kompiuteryje startuoja programą, kurios žinių tikrinimas buvo pasirinktas, bei užkrauna žinioms tikrinti paruoštą dokumentą. Kompiuterio vaizduoklio apačioje užsikrauna „MEDIPlus“ valdymo juosta su funkciniais mygtukais, kurioje yra pateikiami klausimai užduotims vykdyti. Gautą užduotį reikia atlikti programos pagalba, o atsakymo užfiksavimui paspausti funkcinį mygtuką, tada yra pateikiamas sekantis klausimas, kurio užduotį vėl reikia įvykdyti. Užduočių atsakymai yra tikrinami automatiškai, valdymo juostoje rodomas teisingai atsakytų klausimų skaičius. Baigus testą galima pasirinkti rezultatų peržiūrą, kurioje pateikiamas bendras rezultatas, bei neteisingai atliktų klausimų paaiškinimai. Testavimo dokumentai užkraunami iš „MEDIPlus“ sistemos, tačiau jie nėra generuojami, o tik pateikiami atsitiktiniu būdu iš turimų sąrašo.



1 pav. „MEDIAPLUS“ sistemos uždavinių atlikimas

Programų sistema yra labai specifinė ir jos sprendimų pritaikyti ar panaudoti kitoms sistemoms neįmanoma.

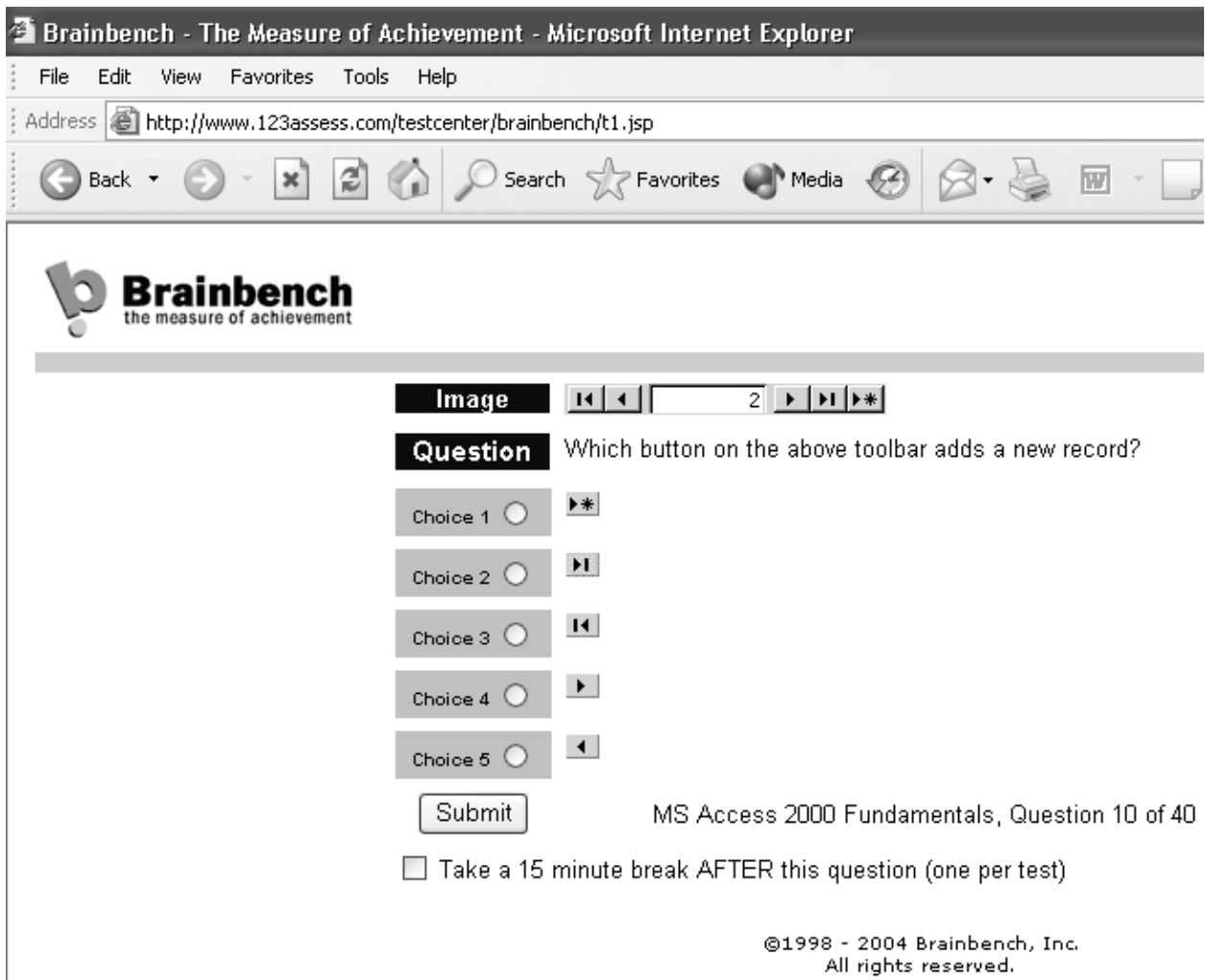
Lietuvos ECDL centro programų sistema skirta kompiuterinio raštingumo testavimui ir ECDL sertifikavimui [8]. Ji sukurta Lotus Notes ir JavaScript technologijų pagrindu. Uždaviniai šioje sistemoje nėra generuojami. Kiekvieną kartą atliekant užduotis jos ir jų eiliškumas lieka tas pats. Tačiau sistemos uždavinių bazė yra nuolat pildoma naujais uždaviniais, pakeičiant jais senus. Uždaviniai apima visas kompiuterių bei programinės įrangos teorinių ir praktinių žinių tikrinimo sritis. Užduočių pateikimo formos yra įvairios: klausimai su atsakymų variantais, paveikslukai su atsakymų variantais, o taip pat klausimai su paveikslukais, kuriuose reikia pele pažymėti aktyvų lauką kaip atsakymą (2 pav.). Tai suteikia vartotojui didesnę veiksmų laisvę, kadangi kartais galimas vienas iš kelių teisingų variantų. Tačiau šioje sistemoje nėra interaktyvių užduočių, bei pateikiamų užduočių generavimo.



2 pav. Lietuvos ECDL testavimo įrankis

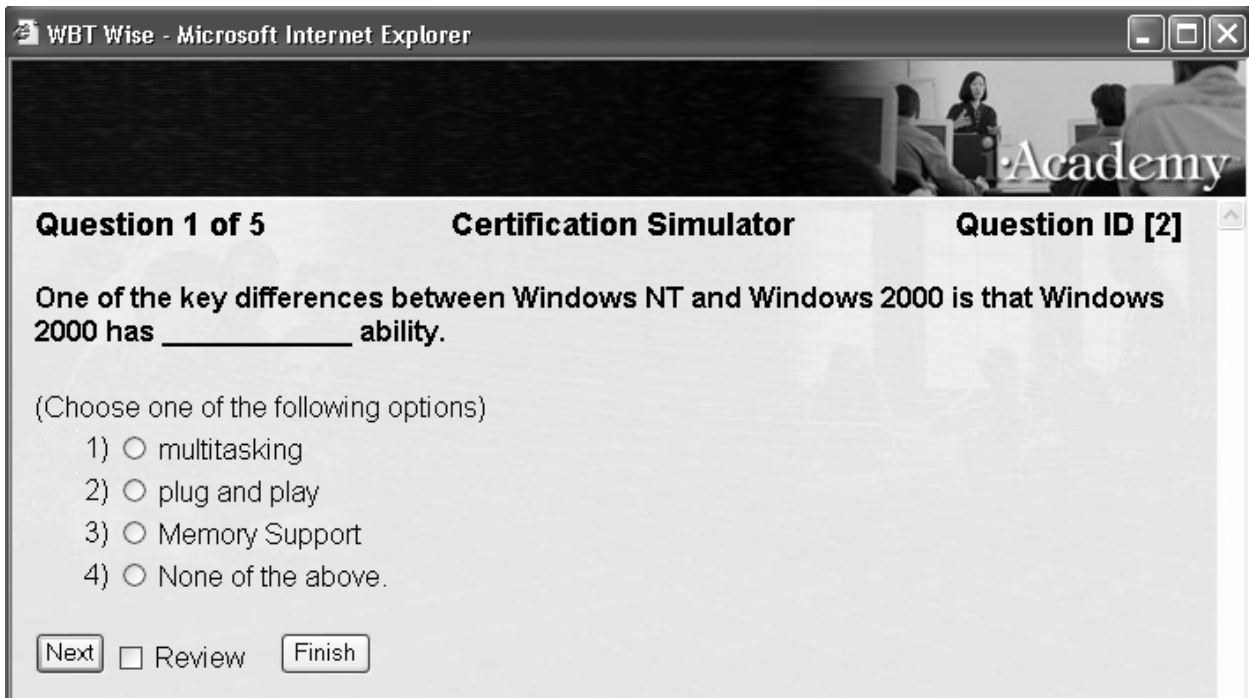
Iš šioje sistemoje realizuotų sprendimų sudomino JavaScript pagalba įgyvendintas aktyvaus lauko, kaip teisingo atsakymo pažymėjimas.

Brainbench – smarkiai išpopuliarėjusi pasaulyje įvairių sričių žinių ir įgūdžių vertinimo sistema [11]. Užduotys sistemoje realizuotos pateikiant klausimą ir galimus atsakymų variantus, taip pat yra ir klausimų su paveikslėliais kartu pateikiant ir atsakymų variantus. Klausimui atsakyti yra skiriamas laiko limitas per kurį reikia pasirinkti atsakymą. Sistema klausimus iš didelės duomenų bazės atsitiktiniu būdu sugeneravusi pateikia vartotojui, taip pat kiekvienam klausimui atsitiktiniu būdu yra generuojamas atsakymų pateikimo eiliškumas, automatinis atsakymų tikrinimas testo pabaigoje pateikia įvertinimą. Klausimų pateikimas realizuotas XML, JSP, JavaScript priemonėmis.



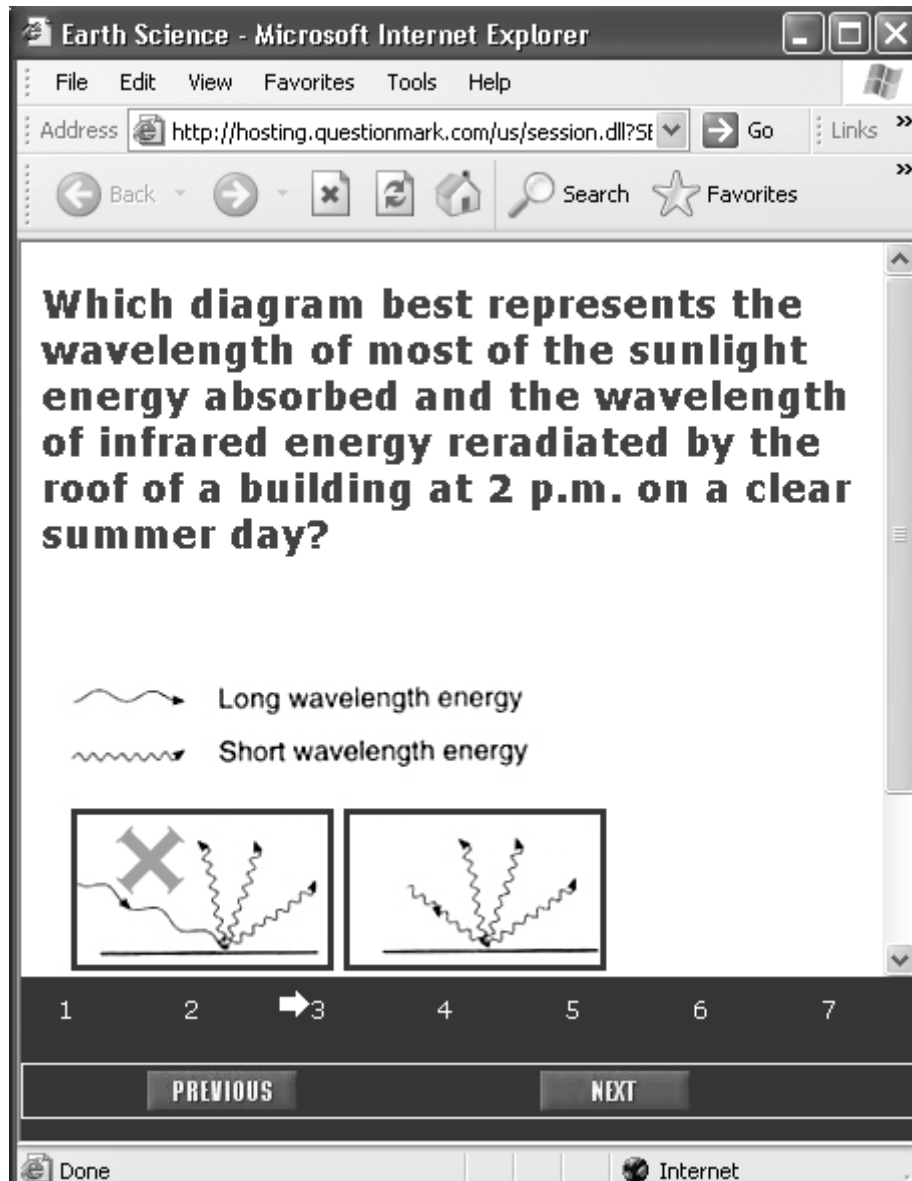
3 pav. „Brainbench“ sistemos vaizdas.

Firmos „WBTwise“ programa, kuri leidžia vartotojui išklausti IT kursą ir po to pabandyti įvertinti savo įgytas žinias [12]. Į kursą įtraukta garsinė, vaizdinė ir tekstinė interpretacija. Tai palengvina vartotojui lengviau įsisavinti informaciją. Šioje sistemoje užduotys yra tik vieno tipo – klausimai su atsakymų variantais (4 pav.). Užduotys pateikiamos atsitiktinai generuojant jų eiliškumą, atsakymų variantai pateikiami lygiai tokiu pačiu principu. Atsakyti klausimai automatiškai tikrinami įvertinant jų teisingumą, o testo pabaigoje pateikiamas galutinis apibendrintas rezultatas.



4 pav. „WBTwise“ testavimo vaizdas

„QuestionMark“ programų sistemoje realizuotas biologijos, chemijos, matematikos, geografijos ir kt. mokslų žinių tikrinimas [13]. Čia pateikiamos užduotys yra įvairių tipų, tai klausimai su atsakymų variantais, klausimai su paveikslukais ir atsakymų variantais, kai kurių klausimų atsakymai su paveikslukais, klausimai kuriuose atsakymą reikia įrašyti į atsakymo lauką, o taip pat klausimai, kuriuose atsakymo paveikslėlyje reikia aktyvų kryžiuką „nutempti“ ant teisingo atsakymo, arba kelis aktyvius atsakymus „nutempti“ į jiems skirtas vietas atsakymo paveikslėlyje. Nors šioje sistemoje užduočių tipų įvairovė nemaža, tačiau užduotys nėra generuojamos ir kiekvieną kart vykdam testą yra pateikiamos tos pačios užduotys.



5 pav. „QuestionMark“ viena iš užduočių.

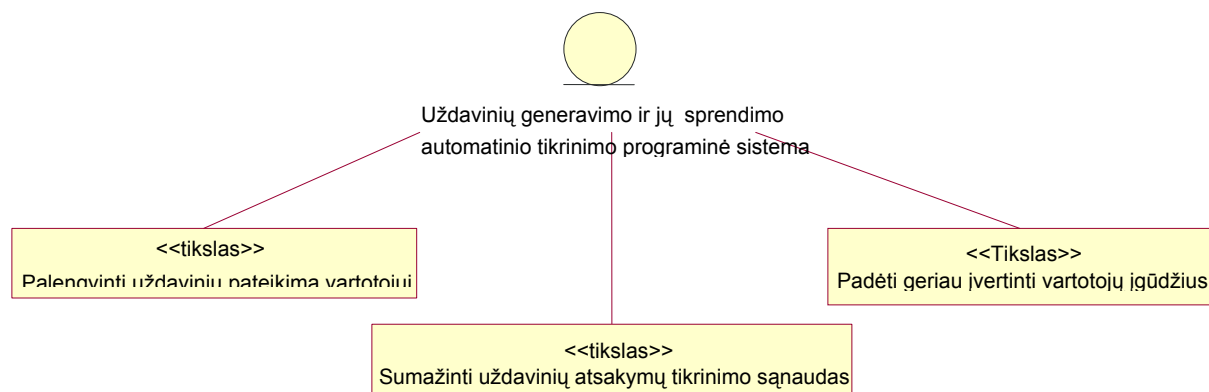
„The Teacher Training Agency“ programa (reikia atsisiųsti ir įdiegti), kuri realizuoja interaktyvų kompiuterinių įgūdžių tikrinimą, tačiau pateikiamos užduotys nėra generuojamos [14].

2.4 Priimto sprendimo pagrindimas

Atlikus analizę buvo nuspręsta realizuoti PS, kuri išspręstų uždavinių generavimo problemą, pateiktą vartotojams interaktyvias užduotis, automatiškai tikrintų rezultatus. Tam buvo suformuluoti sistemos tikslai [1 schema].

2.4.1 Priimto sprendimo tikslai

- ✓ Sistema turi palengvinti (pagerinti) uždavinių pateikimą vartotojams generuodama jiems pateikiamas užduotis, išvengiant monotoniškumo ir uždavinių pastovaus kartojimosi;
- ✓ PS turi iki minimumo sumažinti uždavinių atsakymų tikrinimo sąnaudas pati automatiškai tikrindama sprendžiamų uždavinių atsakymų teisingumą;
- ✓ Sistema, uždavinių sprendimui, vartotojui turi suteikti maksimalią laisvę jo veiksmams, užtikrindama kokybišką žinių tikrinimą.



1 schema. Uždavinių generavimo ir jų sprendimo automatinio tikrinimo programinės sistemos tikslų modelis.

2.4.2 Uždavinių generavimas

Naudojant uždavinių generavimą būtų efektyvesnis vartotojo žinių tikrinimas, išvengiant pateikiamų uždavinių monotoniškumo, galimo atsakymų „nutekėjimo“ tarp vartotojų. Uždaviniai turi būti generuojami kiekvienam testui, kiekvieną kartą. Testą sudaro klausimų grupės, pateikiama bus po vieną klausimą iš kiekvienos grupės. Kiekvienas klausimas turės teisingo/ų ir

klaidingų atsakymų aibę, bei klausimo veiksmą ir objektą. Pvz.: „Kurią programą paleisite (veiksmas) norėdami nusipiešti paveiksluką (objektas)?“ Šiuo atveju teisingų atsakymų aibėje gali būti bet kuri grafinė programa, o klaidingų atsakymų aibėje – bet kuri kita programa. Keičiant klausimo objektą gausime tokio paties tipo užduotis. Vadinasi sudarę objektų aibę, galime generuoti užduotis. Nuo klausimo objekto priklausys teisingi/as atsakymai/as. Atsitiktinai sugeneravus užduoties objektą, turėdami teisingų atsakymų aibę, priklausomai nuo užduoties tipo, kiek teisingų atsakymų pateiksime, galime varijuoti generuodami klaidingus atsakymus.

Interaktyvių uždavinių atveju klausimų sritis bus susieta su užduoties atlikimo lauku, kuriame bus atsitiktinai sugeneruojamas objektų skaičius, iš kurių vienas ar keli priklausomai nuo klausimo veiksmo bus teisingi, o likę atsitiktinai bus generuojami iš klaidingų aibės.

Taigi tokiu principu generuojami uždaviniai padėtų išspręsti užduočių pateikimo problemą.

2.4.3 Rezultatų automatinis tikrinimas

Automatinis rezultatų tikrinimas iš karto leistų sistemos vartotojui sužinoti tikrintų žinių įvertinimą, nereikalautų žmogaus pastangų rezultatų apdorojimui, o pateiktas klaidingai atsakytų klausimų sąrašas, nurodant teisingą atsakymą, leistų vartotojui įvertinti padarytas klaidas.

2.4.4 Užduočių interaktyvumas

Tam, kad užtikrinti įgūdžių tikrinimo kokybiškumą ir suteikti vartotojui daugiau laisvės bus realizuotos interaktyvios praktinių įgūdžių tikrinimo užduotys. Atliekant šias užduotis vartotojo kompiuteryje programų sistema užkrauna realią programą (pvz. explorer) su užduotimi (pvz. Iš katalogo „Temp“ pašalinkite visas tekstines bylas.). Ir šiuo atveju vartotojas turi panaudodamas realius programinės įrangos veiksmus ar komandas atlikti užduotį, kurią dažnai galima išspręsti keliais budais. Taip yra nevaržoma vartotojo veiksmų laisvė ir jis gali pasirinkti sau priimtina sprendimo kelią.

2.5 Projektavimo metodologijos ir technologijų analizė

2.5.1 Servletai

Šiam projektui realizuoti buvo pasirinktas **Apache TomCat** servletų konteineris, kuris palaiko Java servletus.

Apache realizuoja daugumą reikalingų savybių: pranešimus apie klaidas ir problemas, virtualių serverių savybę, autentifikacijos duomenų valdymą. Autentifikacijos duomenų

valdymas leidžia nesulėtinant serverio kurti slaptažodžiu apsaugotus interneto puslapius su dideliu vartotojų skaičiumi.

Apache buvo visapusiškai patikrinta tiek kūrėjų, tiek vartotojų. Apache Grupė nustato griežtus standartus prieš išleisdama naujas serverio versijas. Aptikus klaidas, serverio versija yra tuoj pat taisoma ir kūrėjai kuo greičiau leidžia naują patobulintą Apache serverio versiją. Negalvojama nutraukti bendradarbiavimo su vartotojais. Taip pat ir toliau bus atsižvelgiama į didelių apimčių interneto paslaugų tiekėjus bei į pavienius vartotojus.

2.5.2 Java ir Java Beans

Programų sistemos realizavimui pasirinkta objektinio programavimo kalba Java.

JavaBeans klasės - tai standartiniai objektai, kurie susideda iš:

- klasės kintamųjų;
- metodų *setXxx()* ir *getXxx()* kiekvieno kintamojo reikšmei gauti ir nustatyti;
- konstruktoriaus be parametrų.

Naudojant šias klases, sumažėja veiklos logikos perduodant duomenis tarp komponentų, kurių vienas yra duomenis laikantis, o kitas tuos duomenis apdorojantis komponentas. Tačiau panaudojant šį architektūrinį modelį, vis tiek išlieka skaičiavimų ir duomenų atskyrimo problema.

2.5.3 DBVS

Iš didelio galimų duomenų bazių kiekio:

1 lentelė

Duomenų bazių sąrašas

| | | |
|----------|-----------|------------|
| Adabas D | Ingres | Oracle |
| Informix | ODBC | Unix dbm |
| FilePro | MSQL | Solid |
| Empress | FrontBase | PostgreSQL |
| IBM DB2 | MySQL | Velocis |
| Dbase | InterBase | Ovrimos |

buvo pasirinkta **IBM DB2**. Kaip patikima ir gerai suderinama su JAVA programavimo aplinka. Į IBM DB2 įeina:

- Duomenų žodyno palaikymas
- Daugialygis duomenų saugumo užtikrinimas.

- Priemonės užtikrinančios duomenų integralumą.
- Įvairių kalbų naudojimo palaikymas.
- Vartotojų administravimas, teisių, privilegijų suteikimas.
- Duomenų bazių resursų valdymas
- Priėjimui prie duomenų naudojamos SQL užklausa.

2.5.4 CASE įrankiai

Darbe naudojamoms UML diagramoms projektuoti, kurti ir analizuoti bus pasitelktos šiuolaikinės CASE priemonės - paketas “Rational Rose” bei “Microsoft VISIO” palengvino sistemos analizę.

Automatizuotas programinės įrangos projektavimas (CASE) yra vienas iš metodų, užtikrinančių vartotojų poreikių analizės ir specifikavimo etapo kokybiškumą. CASE priemonėmis iš painaus programinio kodo pereiti į lygmenį, kur architektūra ir projektavimas tampa vaizdingesni, aiškesni ir modifikuojami. Toks PS kūrimo procesas artimesnis vartotojui. CASE priemonės naudojamos visame sistemos kūrimo cikle. Jomis organizuojamas ir valdomas programinės įrangos kūrimas ir ypač jos svarbios kuriant didelius ir sudėtingus projektus. CASE priemonės padeda sistematinti sistemos kūrimo procesą, griežčiau jį kontroliuoti, tai šiuo metu yra aktualu. CASE įrankiai siūlo įvairius programinės įrangos kūrimo metodus. Dalis priemonių pagrįsta struktūriniu metodu, bet pastaruoju metu daugiau orientuojamasi į objektiškai orientuotą programinės įrangos kūrimą.

Projektuojant sistemą bus naudojami UML modeliai:

- Sistemos tikslų modelis
- Paskirstymo diagrama
- Veiklos konteksto diagrama
- Panaudojimo atvejų modelio diagrama
- Duomenų modelis
- Paketų diagrama
- Klasių diagrama
- Procesų veiklos diagrama
- Sekų diagramos
- Komponentų diagrama.

3. Projektinė dalis

3.1 Tikslas

Šio dokumento paskirtis yra įvairiais aspektais išnagrinėti interaktyvių uždavinių generavimo ir jų sprendimo automatinio tikrinimo programų sistemos architektūros vaizdus. Dokumento pagalba bus galima įvertinti pasirinkto architektūrinio sprendimo įtaką kuriamai sistemai.

3.2 Veiklos sfera

Šiame dokumente aprašyti architektūriniai sprendimai tiesiogiai įtakos Uždavinių generavimo ir jų sprendimo automatinio tikrinimo PS, kadangi jais bus remiamasi kuriant produktą. Be to atsiras galimybė pažvelgti į kuriamos sistemos architektūrinio sprendimo privalumus ir trūkumus.

3.3 Bendra apžvalga

Šiame dokumente bus nagrinėjami sekantys punktai:

Architektūrinis vaizdas. Čia bus aprašomas pasirinktas sistemos architektūrinis atvaizdavimas, apžvelgiami skirtingi vaizdai į PS.

Architektūriniai tikslai ir apribojimai. Pateikiami kuriamai sistemai keliami tikslai ir apribojimai, pasirenkamų sprendimų galimybės, įrankių panaudojamumas, panaudojamų komercinių produktų privalumai ir trūkumai.

Panaudojimo atvejų vaizdas. Aprašomi sistemos aktoriai ir jų veiksmi.

Loginis vaizdas. Čia bus pateikta sistemos klasių diagrama ir sistemos objektų sąveikos.

Proceso vaizdas. Pateikiamos objektų bendradarbiavimo ir scenarijų diagramos.

Paskirstymo vaizdas. Pateikiama PS paskirstymo diagrama.

Realizacijos vaizdas. Pateikiama komponentų diagrama.

Duomenų vaizdas. Pateikiama duomenų bazės diagrama.

Dydis ir našumas. Analizuojama kokią įtaką architektūrai turės PĮ našumas ir dydis.

3.4 Architektūrinis vaizdas

Sistemos architektūrai pateikti naudojami penki vaizdai: panaudojimo atvejų, proceso, loginis, paskirstymo ir komponentų.

Panaudojimo atvejų vaizde sudaroma panaudojimo atvejų diagrama bei pateikiamos pagrindinių panaudojimo atvejų specifikacijos.

Loginiam vaizde pateikiama visos dalykinės srities klasių diagrama, kuri bus pagrindu projektuojant sistemos duomenų bazę.

Proceso vaizde pateikiamos svarbiausių klasių objektų sekų, būsenų, bendradarbiavimo, veiklos diagramos.

Paskirstymo vaizde pateikiami aparatūriniai komponentai, kurie sudarys kuriamą PS.

Komponentų vaizde pateiktoje diagramoje parodyti sistemą sudarysiantys programiniai komponentai.

3.5 Architektūriniai tikslai ir apribojimai

Architektūrinius apribojimus lemia pasirinktos sistemos kūrimo priemonės: Java objektinio programavimo kalba, Java appletai, JSP dinaminiai puslapiai integruojantys Java, Tomcat servletų konteineris, IBM DB2 serveris duomenims saugoti.

3.6 Panaudojimo atvejų vaizdas

3.6.1 Panaudojimo atvejų realizacijos

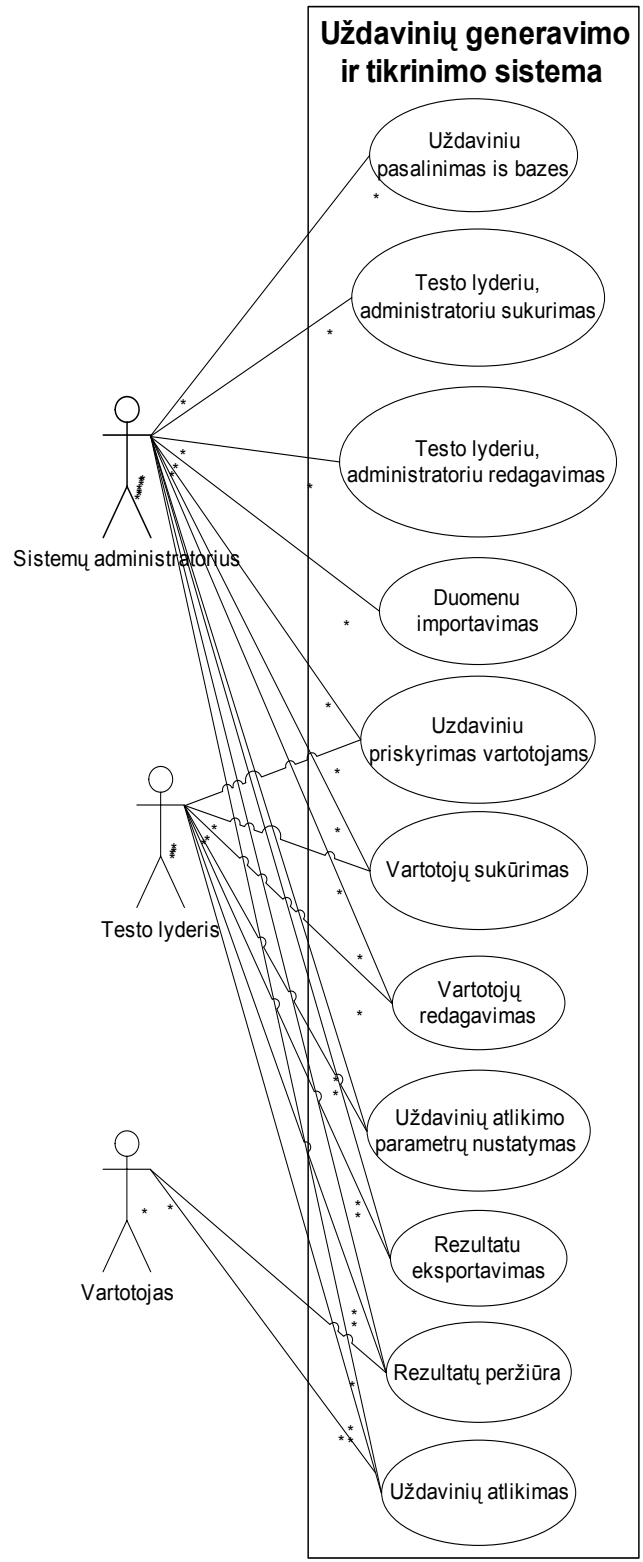
Aktoriai:

- sistemos administratorius,
- testo lyderis,
- vartotojas (uždavinių sprendėjas).

Panaudojimo atvejai:

- uždavinių šalinimas iš bazės,
- administratorių, testo lyderių sukūrimas,

- administratorių, testo lyderių redagavimas,
- duomenų importavimas,
- vartotojų sukūrimas,
- vartotojų redagavimas,
- uždavinių priskyrimas vartotojams,
- uždavinių atlikimo parametrų nustatymas,
- uždavinių atlikimas,
- rezultatų peržiūra,
- rezultatų eksportavimas.



2 schema. Veiklos panaudojimo atvejų modelis

1. Panaudojimo atvejis uždavinių pašalinimas iš bazės

Aprašas: esamų uždavinių pašalinimas iš bazės. Vartotojui išvedami esamų uždavinių sąrašas.

Vartotojas/Aktorius: sistemų administratorius.

Prieš sąlyga: prisijungęs sistemų administratorius.

Sužadinimo sąlyga: aktyvuojamas meniu punktas.

Po-sąlyga: atnaujinama uždavinių bazė.

2. Panaudojimo atvejis testo lyderių, administratorių sukūrimas

Aprašas: sukurti naują testo lyderį, administratorių.

Vartotojas/Aktorius: sistemų administratorius.

Prieš sąlyga: prisijungęs sistemų administratorius.

Sužadinimo sąlyga: aktyvuojamas meniu punktas.

Po-sąlyga: duomenys įtraukiami į duomenų bazę.

3. Panaudojimo atvejis testo lyderių, administratorių redagavimas

Aprašas: pakeisti testo lyderių, administratorių duomenis, pašalinti testo lyderius, administratorius

Vartotojas/Aktorius: sistemų administratorius

Prieš sąlyga: prisijungęs sistemų administratorius, egzistuoja jo sukurti testo lyderiai ir/arba administratoriai.

Sužadinimo sąlyga: pasirenkamas testo lyderis arba administratorius, aktyvuojamas atitinkamas meniu punktas.

Po-sąlyga: duomenys atnaujinami DB.

4. Panaudojimo atvejis duomenų importavimas

Aprašas: importuoti vartotojus, uždavinius.

Vartotojas/Aktorius: sistemų administratorius.

Prieš sąlyga: prisijungęs sistemų administratorius.

Sužadinimo sąlyga: pasirenkamas meniu punktas.

Po-sąlyga: duomenys įrašomi į DB.

5. Panaudojimo atvejis uždavinių priskyrimas vartotojams

Aprašas: aukščiausio hierarchijos lygio administratorius gali priskirti visus arba dalį uždavinių žemesnio lygio administratoriams, kurie tada gali priskirti žemesniems vartotojams uždavinius iš jiems priskirtos aibės.

Vartotojas/Aktorius: sistemų administratorius, testo lyderis.

Prieš sąlyga: prisijungęs sistemų administratorius, testo lyderis ir egzistuoja bent vienas žemesnio lygio vartotojas.

Sužadinimo sąlyga: pasirenkamas vartotojas, sužadinamas meniu punktas.

Po-sąlyga: duomenys atnaujinami DB.

6 Panaudojimo atvejis vartotojų sukūrimas

Aprašas: sukurti naują vartotoją(us).

Vartotojas/Aktorius: sistemų administratorius, testo lyderis.

Prieš sąlyga: prisijungęs administratorius arba testo lyderis.

Sužadinimo sąlyga: aktyvuojamas atitinkamas meniu punktas.

Po-sąlyga: duomenys įrašomi į DB.

7. Panaudojimo atvejis vartotojų redagavimas

Aprašas: vartotojo informaciją, pašalinti vartotoją.

Vartotojas/Aktorius: sistemų administratorius, testo lyderis.

Prieš sąlyga: prisijungęs sistemų administratorius arba testo lyderis, sukurtas bent vienas vartotojas.

Sužadinimo sąlyga: pasirenkamas vartotojas iš vartotojų sąrašo.

Po-sąlyga: duomenys atnaujinami DB.

8. Panaudojimo atvejis uždavinių atlikimo parametrų nustatymas.

Aprašas: nustatyti uždavinių atlikimo parametrus (kiek kartų galima atlikti testą, nuo kokios datos iki kokios galima atlikti testą, laiko intervalas kada vel galima atlikti ta pati testa) žemesnio lygio vartotojams.

Vartotojas/Aktorius: sistemų administratorius, testo lyderis.

Prieš sąlyga: prisijungęs sistemų administratorius arba testo lyderis.

Sužadinimo sąlyga: pasirenkamas vartotojas, aktyvuojamas atitinkamas meniu punktas.

Po-sąlyga: duomenys atnaujinami DB.

9. Panaudojimo atvejis rezultatų peržiūra

Aprašas: peržiūrėti, nagrinėti atliktų uždavinių rezultatus.

Vartotojas/Aktorius: sistemų administratorius, testo lyderis, vartotojas.

Prieš sąlyga: prisijungęs administratorius, testo lyderis arba vartotojas; yra užfiksuoti uždavinių rezultatai.

Sužadinimo sąlyga: aktyvuojamas atitinkamas meniu punktas.

Po-sąlyga: atvaizduojami duomenys.

10. Panaudojimo atvejis rezultatų eksportavimas

Aprašas: vartotojų atliktų užduočių rezultatų eksportavimas.

Vartotojas/Aktorius: administratorius, testo lyderis.

Prieš sąlyga: prisijungęs administratorius arba testo lyderis; yra užfiksuoti uždavinių rezultatai.

Sužadinimo sąlyga: aktyvuojamas atitinkamas meniu punktas.

Po-sąlyga: duomenys paimami iš DB ir išsaugomi txt byloje.

11. Panaudojimo atvejis uždavinių atlikimas

Aprašas: pasirinkto tipo uždavinių atlikimas.

Vartotojas/Aktorius: administratorius, testo lyderis, vartotojas.

Prieš sąlyga: prisijungęs administratorius, testo lyderis arba vartotojas, jeigu yra suteikta teisė atlikti užduotis; prisijungusiam vartotojui yra priskirti uždaviniai, nesibaigęs uždavinių atlikimo limitas.

Sužadinimo sąlyga: aktyvuojamas atitinkamas meniu punktas.

Po-sąlyga: atliktų uždavinių rezultatai išsaugojami DB.

3.7 Loginis vaizdas

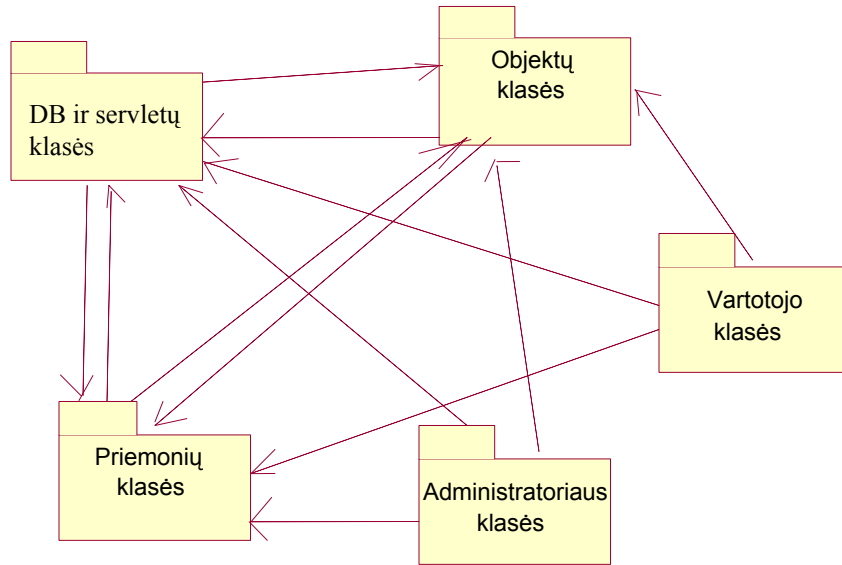
Vartotojo sąsajos modeliai.

3.7.1 Apžvalga

Kuriamos sistemos klasės galima suskirstyti į keletą paketų. Tai:

- DB ir servletų klasės teikiančios duomenų apdorojimo paslaugas, bei ryšį su DB
- Objektų (testo, klausimo, personos, rezultatų) klasės, naudojamos kuriant, pateikiant, atvaizduojant objektus.
- Vartotojo klasės, vykdomos prisijungus vartotojui.
- Administratoriaus klasės, vykdomos prisijungus administratoriui.

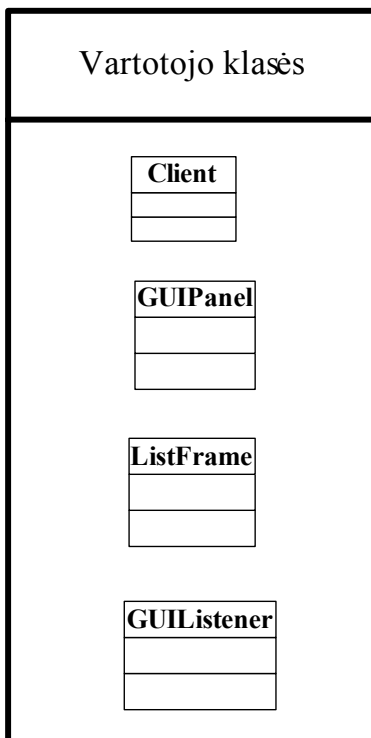
- Priemonių klasės – dialogai, klaidų pranešimai ir kt.



3 schema. **Paketų diagrama**

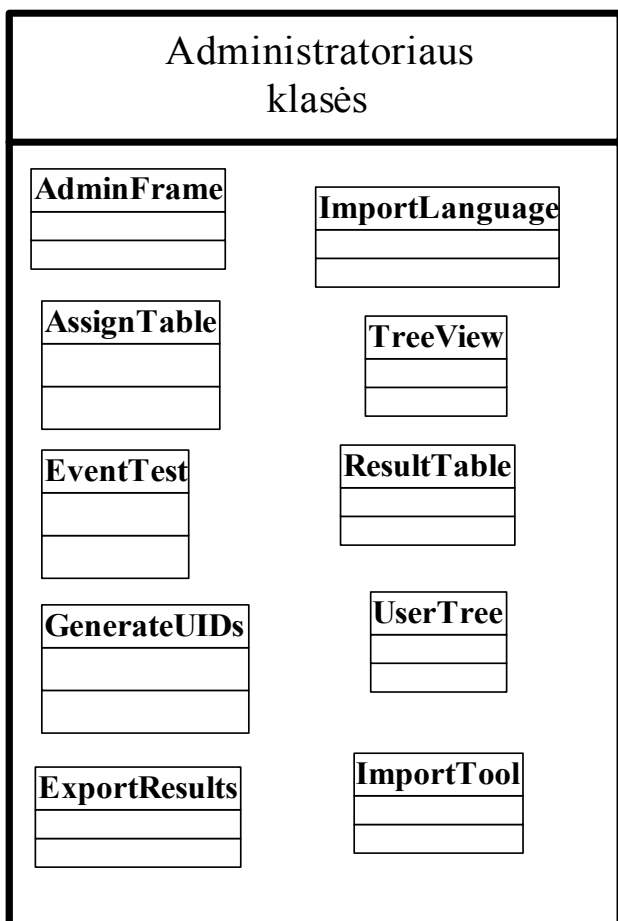
3.7.2 Vartotojo klasės

Vartotojo pagrindinės klasės diagramoje:



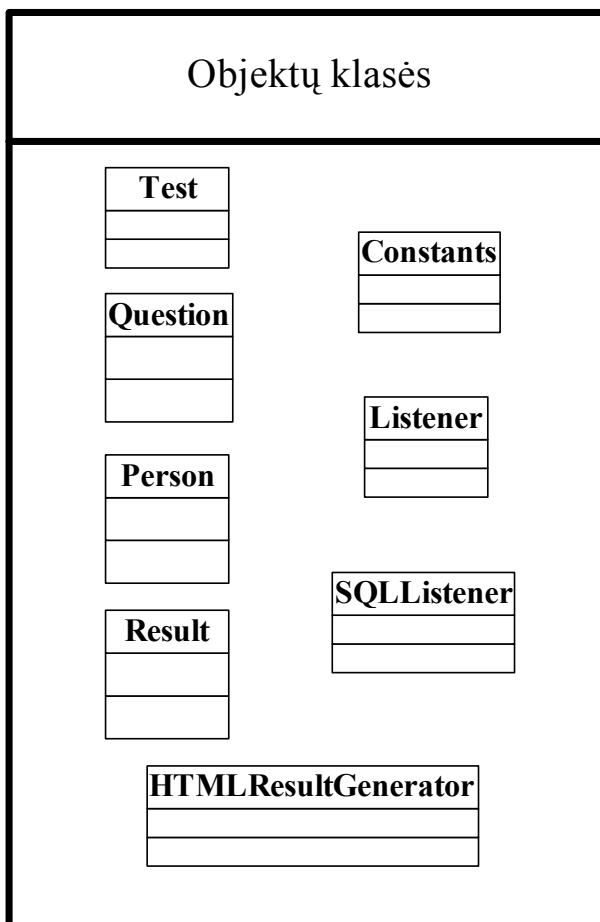
4 schema. **Vartotojo klasės**

3.7.3 Administratoriaus klasės



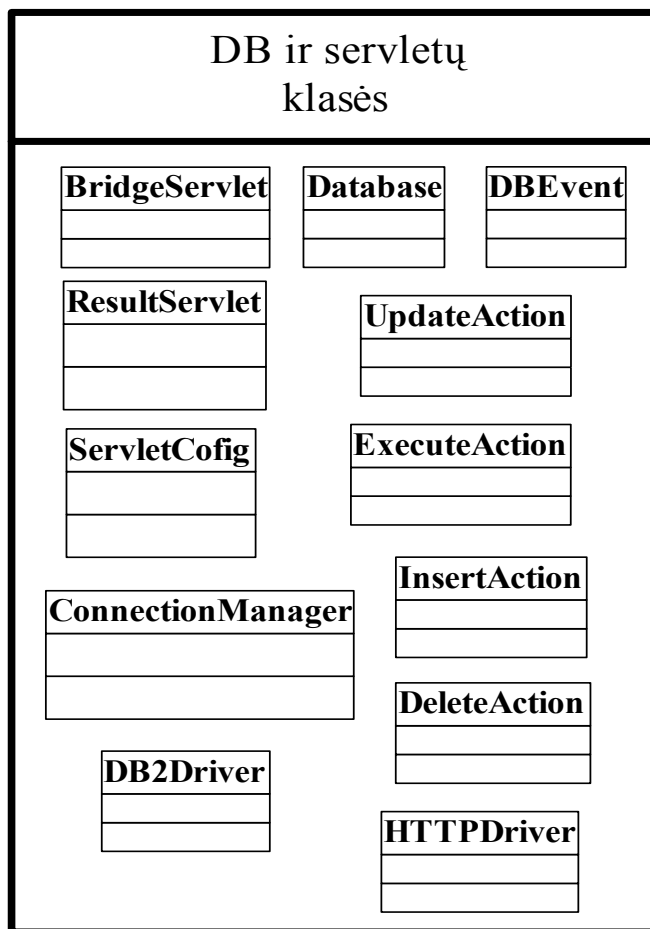
5 schema. Administratoriaus klasės

3.7.4 Objektų klasės



6 schema. **Objektų klasės**

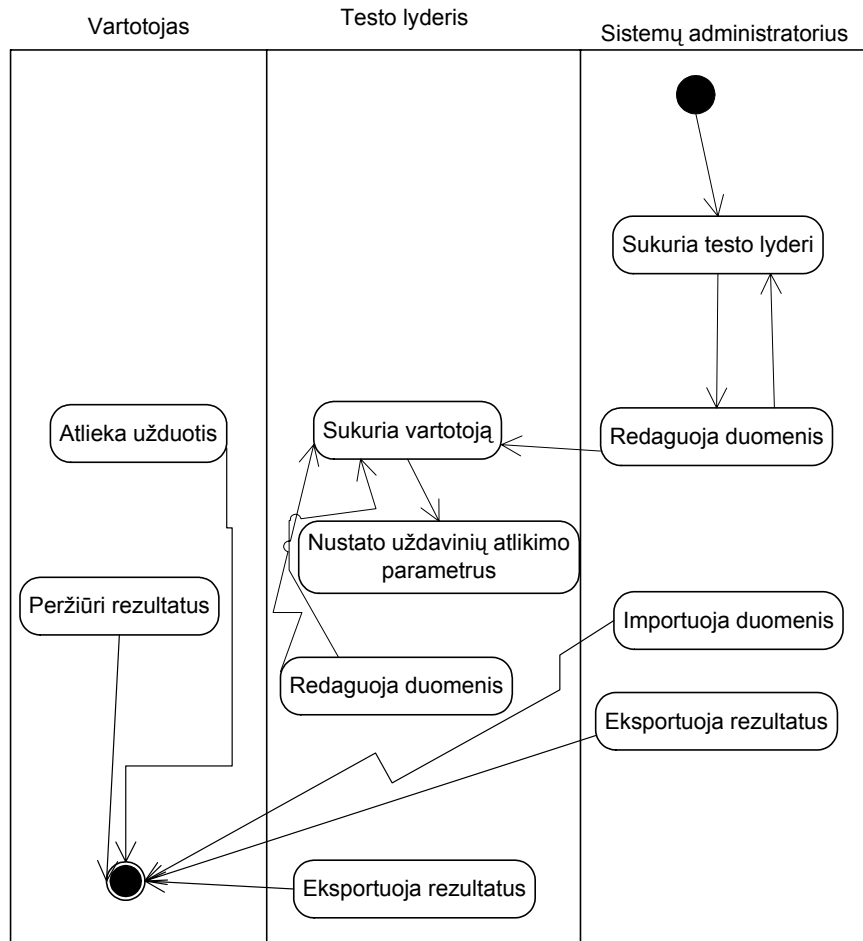
3.7.5 DB ir servletų klasės



7 schema. **DB ir servletų klasės**

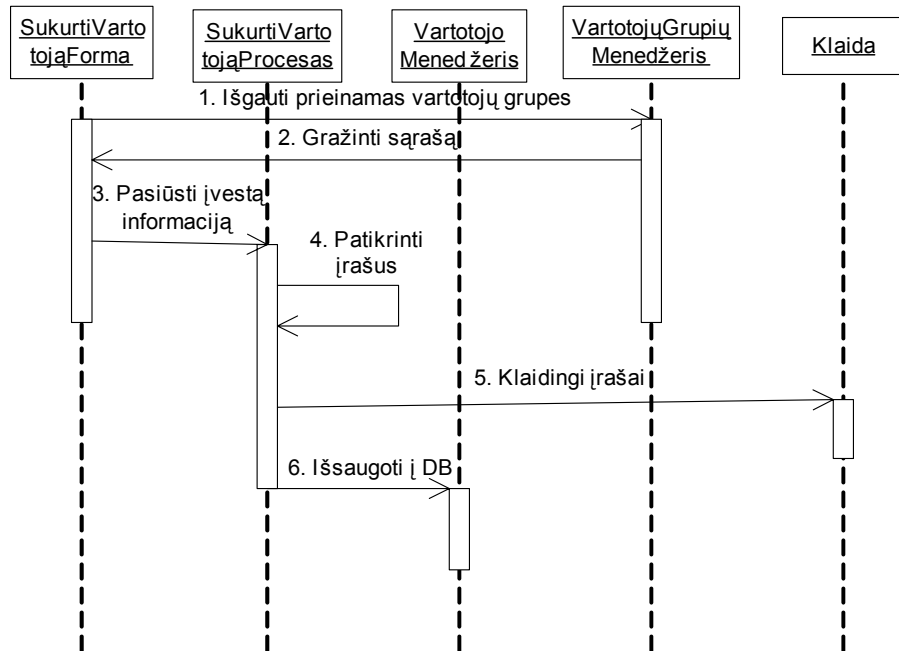
3.8 Proceso vaizdas

3.8.1 Uždavinių generavimo ir jų sprendimo tikrinimo sistemos procesas



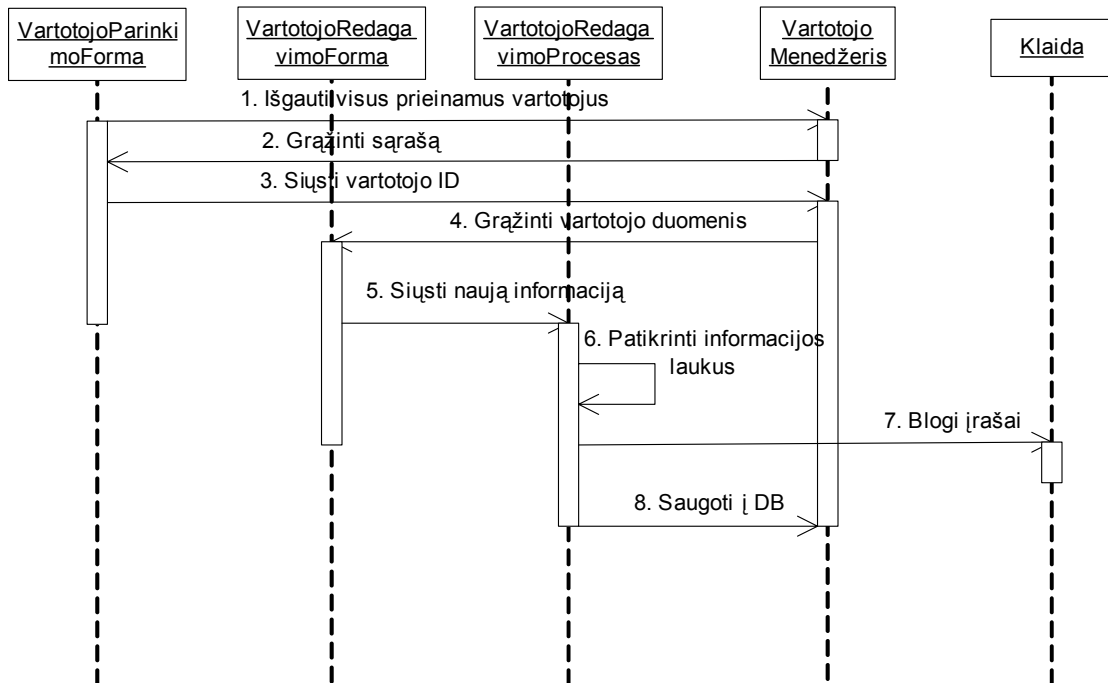
8 schema. **Procesų veiklos diagrama**

3.8.2 Vartotojo kūrimo sekos diagrama



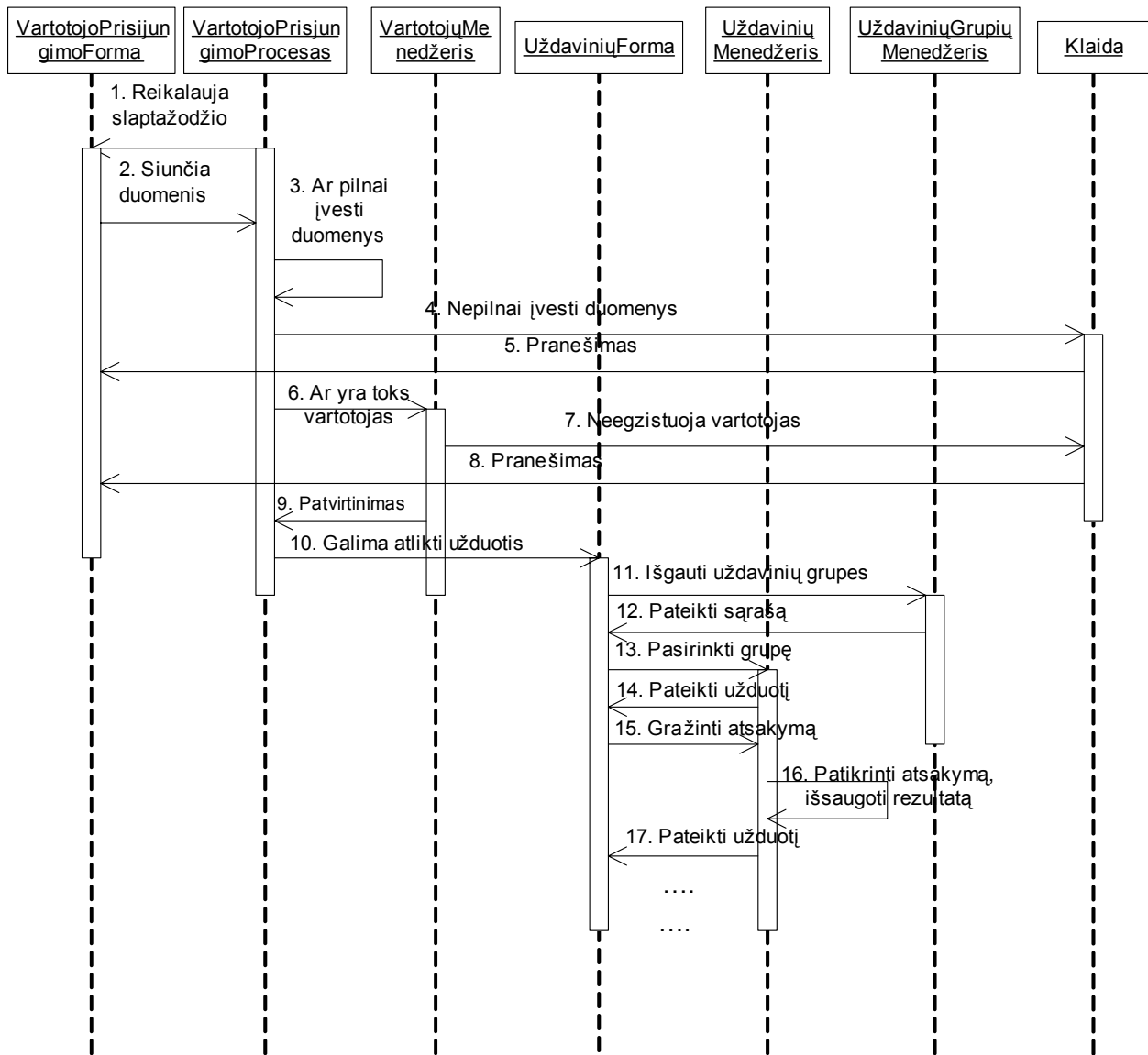
9 schema. Vartotojo kūrimo sekos diagrama

3.8.3 Vartotojo redagavimo sekos diagrama



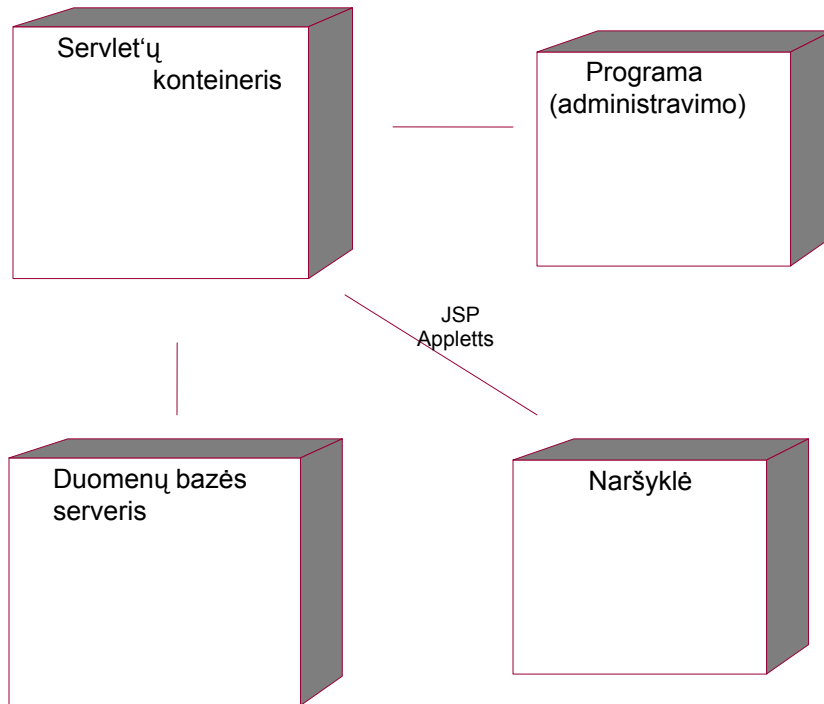
10 schema. Vartotojo redagavimo sekos diagrama

3.8.4 Uždavinio atlikimo sekos diagrama



11 schema. Uždavinio atlikimo sekos diagrama

3.9 Paskirstymo vaizdas



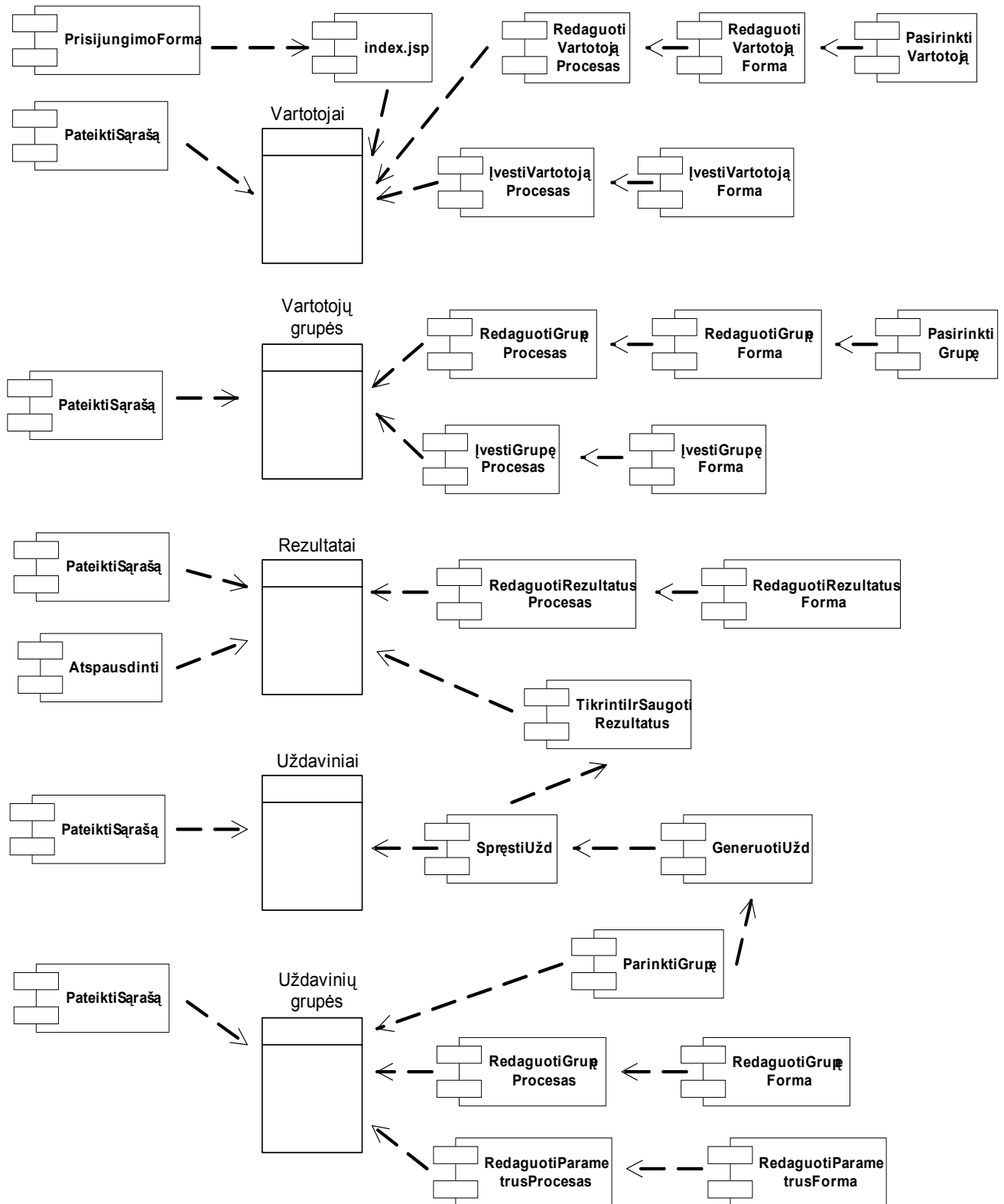
12 schema. Paskirstymo diagrama

3.10 Realizacijos vaizdas

3.10.1 Apžvalga

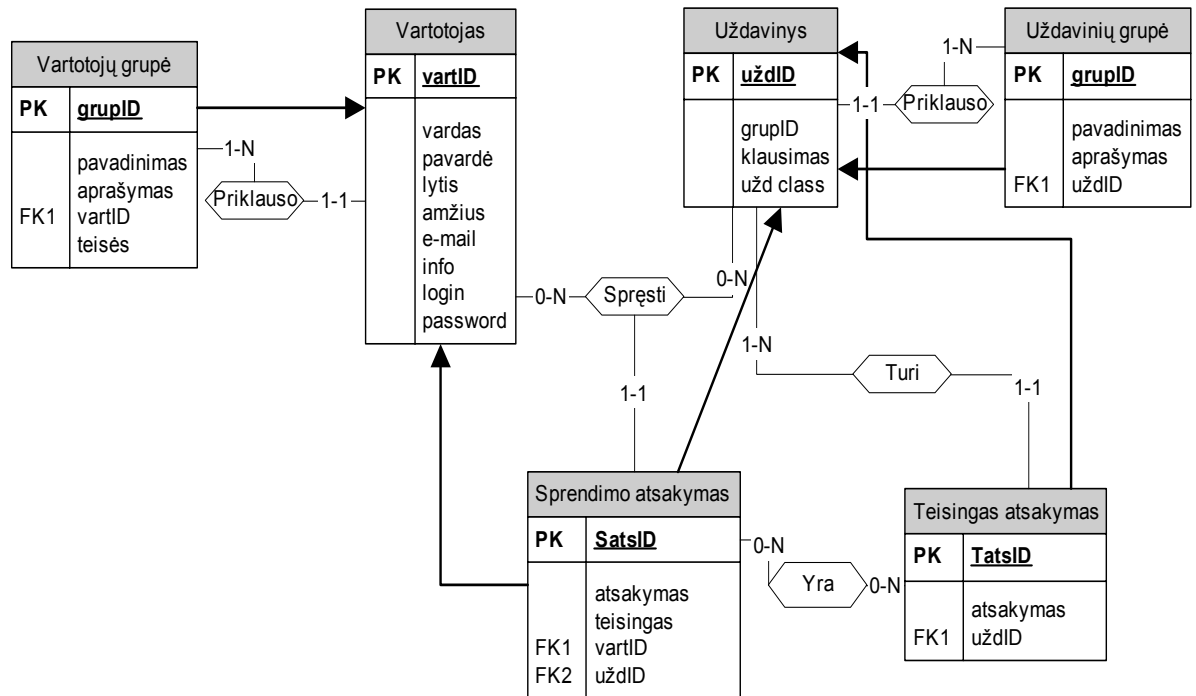
Visumoje sistemą sudarys keletas paketų. Skyriuje 5 pateikiamos klasių paketo struktūros.

3.10.2 Komponentų sąveikos diagrama



13 schema. Komponentų diagrama

3.10.3 Duomenų vaizdas



14 schema. **Sistemos duomenų modelis - ER diagrama.**

3.11 Dydis ir našumas

Kuriama programų sistema yra vidutinio dydžio. Sistemos našumą daugiausia įtakos pasirinktos kūrimo priemonės: Java appletai, servletai, duomenų bazių valdymo sistema IBM DB2.

3.12 Apibendrinimas

Pagal aprašytą PS architektūrą buvo realizuota programinė įranga skirta uždavinių generavimui ir jų sprendimo automatiniam tikrinimui.

4. Tyrimo dalis

4.1 Paskirtis

Tiriamoji dalis skirta realizuotos uždavinių generavimo bei jų sprendimo automatinio tikrinimo programinės įrangos kokybės analizei.

4.2 Realiai atlikto darbo kokybės analizės tikslai

- ✓ Patikrinti ar programų sistema atitinka reikalavimų specifikaciją.
- ✓ Įsitikinti ar programų sistema sukurta pagal standartus.
- ✓ Įvertinti sistemos našumą, patikimumą bei testuojamumą.
- ✓ Įvertinti ar vartotojams lengva naudotis sistema.
- ✓ Įvertinti programų sistemos tobulinimo galimybes.

4.3 Funkcionalumo tyrimas

Tyrimo eigoje programų sistema naudojosi įvairaus lygio vartotojai. Buvo siekiama patikrinti sistemos funkcionalumo atitikimą vartotojo reikalavimams. Vartotojams tyrimo metu buvo pateiktas sistemos funkcijų sąrašas, kuriame jie pažymėjo programų sistemos atitikimą specifikacijai (2 lentelė). Išpildyti reikalavimai pažymėti pliusu, o atskiriems panaudojimo atvejams nenumatyti veiksmai pažymėti minusu.

Aktoriai: A – sistemos administratorius, L – testo lyderis, U – vartotojas (uždavinių atlikėjas).

2 lentelė

Sistemos funkcionalumas

| Reikalavimo Nr. | Funkcionalumas | Aktoriai | | |
|-----------------|------------------------------------|----------|---|---|
| | | A | L | U |
| 1.1 | Prisijungimas prie darbo aplinkos | + | + | + |
| 1.2 | Atsijungimas nuo darbo aplinkos | + | + | + |
| 2 | Darbo aplinkos kalbos pasirinkimas | + | + | + |

| | | | | |
|-------|--|---|---|---|
| 3.1 | Administratorių, testo lyderių sukūrimas | + | - | - |
| 3.2 | Administratorių, testo lyderių redagavimas | + | - | - |
| 4.1 | Naujo vartotojo sukūrimas, redagavimas | + | + | - |
| 5.1 | Uždavinių atlikimo parametrų nustatymas | + | + | - |
| 5.2 | Uždavinių atlikimas | + | + | + |
| 5.3 | Rezultatų peržiūra | + | + | + |
| 5.4.1 | Rezultatų eksportavimas | + | + | - |
| 5.4.2 | Duomenų importavimas | + | - | - |
| 6 | Uždavinių šalinimas iš bazės | + | - | - |

Gauti tyrimo rezultatai patvirtino programų sistemos funkcionalumo atitikimą sistemos specifikacijai.

4.4 Standartų laikymasis

Programų sistemos kūrimo procesas aprašytas projektinėje dalyje. Kuriant PS buvo laikomasi šio dokumento ir jame aprašytų metodikų.

4.5 Sistemos našumas

PS veikimo greitis didele dalimi priklauso nuo Interneto ryšio greičio duomenų bazės serveryje, Tomcat serveryje (jei įdiegtas ne DB serveryje), bei nuo vartotojo ryšio spartos. Papildomai sistemos darbo greitį gali įtakoti duomenų bazės serverio ar Tomcat serverio kitų procesų didelis apkrovimas, jeigu tokie būtų.

Sistemos išnaudojama kietojo disko vieta yra nedidelė ir daug resursų šiuo atžvilgiu nereikalauja. Daugiausia vietos gali pririnkti duomenų bazei, jeigu vartotojų skaičius išaugtų iki labai didelio.

4.6 Sistemos patikimumas

Sistemos patikimumas buvo padidintas ištaius testavimo fazės metu aptiktas kelias klaidas. Daugiau sistemos korektiško veikimo sutrikimų nepasitaikė, todėl šią programų sistemą galima laikyti patikima.

4.7 Sistemos testuojamumas

Kadangi realizuotoje programų sistemoje nėra ypatingai sudėtingų loginių skaičiavimų ar veiksmų, tai atlikti PĮ testavimą nebuvo sudėtinga ir galime teigti, kad sistemos testuojamumas yra lengvas.

4.8 Vartotojo sąsaja

Realizuotos programų sistemos vienas iš nefunkcinių reikalavimų buvo paprasta ir lengvai supranta vartotojo sąsaja. Šis reikalavimas buvo sėkmingai įgyvendintas. Sistemoje realizuoti tik pagrindiniai funkciniai mygtukai, ji neapkrauta papildomais nereikalingais dalykais, paaiškinimai ir pranešimai yra aiškūs ir tikslūs. Vartotojui norint naudotis programų sistema jokių specialių žinių ar įgūdžių įgyti nereikia, pakanka elementarių naudojimosi kompiuterių žinių.

4.9 Sistemos saugumas

Sistemos perduodamų, bei gaunamų duomenų saugumas yra užtikrinamas panaudojant RSA, MD5, DES kodavimo algoritmus, bei UTF-8 koduotę. Be to sistema galima naudotis tik autorizuotai prisijungus.

4.10 Kokybės vertinimo procesas

4.10.1 Interviu su užsakovu

Interviu su užsakovu metu, buvo aptariami reikalingi programų sistemos patobulinimai. Buvo sukurtas reikalingų pakeitimų sąrašas (3 lentelė), kuriame programos kūrėjas atlikęs pataisymus pažymėjo jų įvykdymą.

3 lentelė

Patobulinimų sąrašas

| Nr. | Reikalingi pataisymai | Įvykdymas |
|------------|--|------------------|
| 1 | Didesni pateikiamų uždavinių paveikslėliai. | + |
| 2 | Išvedant rezultatus pateikti neteisingai atliktų uždavinių sąrašą, nurodant vartotojo atsakymą ir teisingą atsakymą. | + |
| 3 | Vartotojo sąsajos „daugiakalbiškumo“ realizavimas. | + |

Visi numatyti programų sistemos patobulinimai skirti sąsajai su vartotoju gerinti – padaryti ją „draugiškesnę“ ir dar daugiau informatyvesnę, siekiant maksimaliai patenkinti vartotojų poreikius. Vartotojo sąsajos kalbos pasirinkimo galimybės realizavimas įgalina sistemos vartojimo rinkos praplėtimą.

4.10.2 Apklausų anketos

Apklausų anketos buvo naudojamos sužinoti vartotojų nuomonę apie realizuotos programų sistemos funkcionalumo atitikimą sistemos specifikacijai. Anketose buvo pateikiami klausimai atspindintys reikalavimuose iškeltas sistemos funkcijas. Jų rezultatas yra pateiktas 2 lentelėje.

4.11 Vertinimo rezultatai

Šio darbo tyrimo dalyje atlikta realizuotos uždavinių generavimo ir jų sprendimo automatinio tikrinimo programų sistemos kokybės analizė. Pateiktas kokybės vertinimo proceso aprašymas. Analizuojant gautus tyrimo metu rezultatus aptartos bei iškeltos sistemos patobulinimų galimybės. Realizuoti patobulinimo uždaviniai – sistemos kokybės padidimui. Projekto kūrimo procesas laikomas pasikartojančiu. Jo našumas teigiamas.

5. Eksperimentinė dalis

5.1 Eksperimento tikslas ir uždaviniai

Šių eksperimentinių tyrimų tikslas – patvirtinti arba paneigti darbo metu suformuotas hipotezes ir teiginius. Galima išskirti svarbiausius uždavinius, kurių atžvilgiu bus vykdomi tyrimai:

- ✓ Sužinoti vartotojų požiūrį apie kompiuterines žinių tikrinimo sistemas ir naudojimosi jomis paplitimą tarp skirtingo amžiaus, profesijos vartotojų;
- ✓ Ištirti vartotojų požiūriu tokios žinių tikrinimo sistemos teikiamas galimybes ir skirtumus nuo kitų panašių programų sistemų;
- ✓ Išsiaiškinti šios žinių tikrinimo sistemos stipriąsias ir silpnąsias puses;

5.2 Eksperimento objektas

Eksperimento metu bus nagrinėjama uždavinių generavimo ir jų sprendimo automatinio tikrinimo programų sistema.

5.3 Eksperimentinio tyrimo eiga

Vykdam tyrimus realizuota programų sistema buvo pateikta naudojimui. Vartotojai buvo apklausiami, bei jiems buvo pateiktos anketos. Eksperimento metu iš viso buvo apklausti ir išanketuoti 30 respondentų, iš kurių 20 – besimokantys, 7 – mokytojai, dėstytojai, 3 – nepriklausantys švietimo sričiai. Pagal amžių respondentai pasiskirstė taip: iki 18m. amžiaus apklausti 2 respondentai, 18-23m. – 11, 24-30m. – 9, 31-40m. – 4, 41-50m. – 3, 51-60m. – 1, 61 ir daugiau – 0. Pagrindinis atrankos kriterijus buvo mokėjimasis naudotis kompiuteriu.

5.3.1 Vartotojų požiūris į kompiuterines žinių tikrinimo sistemas

Atliekant eksperimentą pirmiausiai buvo domimasi respondentų naudojimosi tokio pobūdžio sistemomis lygiu (nesinaudojo, naudojosi bent vieną kartą, naudojasi dažnai). Iš gautų duomenų paaiškėjo (3 priedas), kad daugelis respondentų gana dažnai susiduria su tokio pobūdžio žinių tikrinimo sistemomis, todėl galima daryti išvadą, kad tokie vartotojai gali nusakyti prieš tai naudotų ir eksperimentuojamosios sistemos funkcionavimo trūkumus bei privalumus, netiesiogiai jas lygindami tarpusavyje. Dėl šios priežasties respondentams buvo

pateikiami anketiniai klausimai susiję su tokių sistemų funkcionavimu, t.y. vartotojo darbo lauko, klausimų, rezultatų pateikimu.

5.3.2 Žinių tikrinimo sistemų savybių tyrimas

Sukurta kompiuterinių žinių tikrinimo sistema pasižymi savybe nebūdinga daugeliui tokio pobūdžio sistemų. Ši savybė – tai užduočių generavimas, t.y. pateikiamos užduotys nuolat kinta, neturi įprasto kitoms sistemoms klausimų eiliškumo kartojimosi, keičiasi galimų atsakymų variantai. Atlikus apklausą šią savybę kaip ypač novatorišką ir patrauklią išskyrė asmenys, nuolat tikrinantys besimokančiųjų žinias (4 priedas). Paprašyti motyvuoti, kodėl ši savybė tokia priimtina, daugelis teigė, kad tai ypač sumažintų studentų, moksleivių „nusirašymo“ galimybes, bei pakeltų gebėjimą savarankiškai mąstyti.

Respondentų grupė, kurie įgytas žinias nuolat tikrina panašiose sistemose, išskyrė automatinį sprendimų tikrinimą, kaip šios sistemos privalumą (5 priedas). Didesnė dalis respondentų teigė nori tik atlikę užduotis iš karto sužinoti savo įvertinimą. Tai ypač naudinga tiems, kurie praktikuojasi ir nuolat tikrina savo žinias. Dėl šios priežasties daugelis teigė, jog džiaugtųsi turėdami galimybę pamatyti neteisingai atliktas užduotis, bei sužinoti teisingus jų atsakymus.

Kaip dar vieną šios sistemos privalumą respondentai įvertino pateikiamas realias, interaktyvias užduotis, kurioms atlikti vartotojai turi panaudoti realius programinės įrangos naudojimo veiksmus (4, 5 priedai). Tokios interaktyvios užduotys pagerintų vartotojų sugebėjimą pritaikyti žinias praktikoje. Taip pat taip pateikiami klausimai verčia vartotoją mąstyti, suteikia visišką veiksmų laisvę, o žinių tikrinimą padaro įdomesnį.

Šiuolaikinės panašaus pobūdžio žinių tikrinimo sistemos yra ypač universalios kalbos požiūriu. Todėl buvo teiraujama respondentų ar būtina ir reikalinga tokiai programai galimybė pakeisti vartotojo sąsajos kalbą į kitą (4, 5 priedai). Didesnei daliai respondentų tai atrodė ne ypač reikalingas dalykas, tačiau esant didelei tokių programų paklausai bei pasiūlai, ši savybė padėtų geriau įsitvirtinti užsienio informacinių technologijų rinkoje.

5.4 Eksperimento apibendrinimas

Atlikus eksperimentą ir remiantis gautais duomenimis galima teigti, kad ši sistema pasižymi daugeliu savybių, kurios yra labai patrauklios vartotojams, pagerinančios žinių ir įgūdžių tikrinimą. Programų sistema realizuoja kiekvienos pateikiamos užduoties generavimą,

siekiant išvengti užduočių pastovaus kartojimosi, pateikia interaktyvias užduotis, reikalaujančias realių vartotojų įgūdžių panaudojimo. Suteikia vartotojams galimybę ne tik iš karto sužinoti savo rezultatus, bet ir peržiūrėti neteisingai atliktas užduotis.

6. Išvados

- Uždavinių generavimas ir jų sprendimo automatinis tikrinimas yra labai aktualus šiuolaikinėse žinių vertinimo sistemose.
- Siekiant objektyvaus ir kokybiško žinių įvertinimo stengiamasi, kad pateikiamos užduotys nesikartotų.
- Buvo priimtas sprendimas generuoti kiekvieną pateikiamą užduotį, jos eiliškumą, priklausomai nuo užduoties tipo generuoti uždavinio kintamuosius, uždavinio neteisingų atsakymų variantus, jų eiliškumą. Automatiškai tikrinti užduočių atsakymus, pateikti neteisingai atliktų uždavinių sąrašą.
- Efektyvesniam įgūdžių tikrinimui buvo realizuoti interaktyvūs uždaviniai, kurių sprendimas reikalauja panaudoti realius veiksmus.
- Pasirinktomis programavimo technologijomis (Java, applets, servlets, IBM DB2) buvo realizuota suprojektuota uždavinių generavimo ir jų sprendimo automatinio tikrinimo programų sistema.
- PS kokybės analizės metu iškelti ir realizuoti sistemos patobulinimai.
- Atliktas eksperimentas patvirtino nagrinėtų problemų svarbą.

7. Literatūra

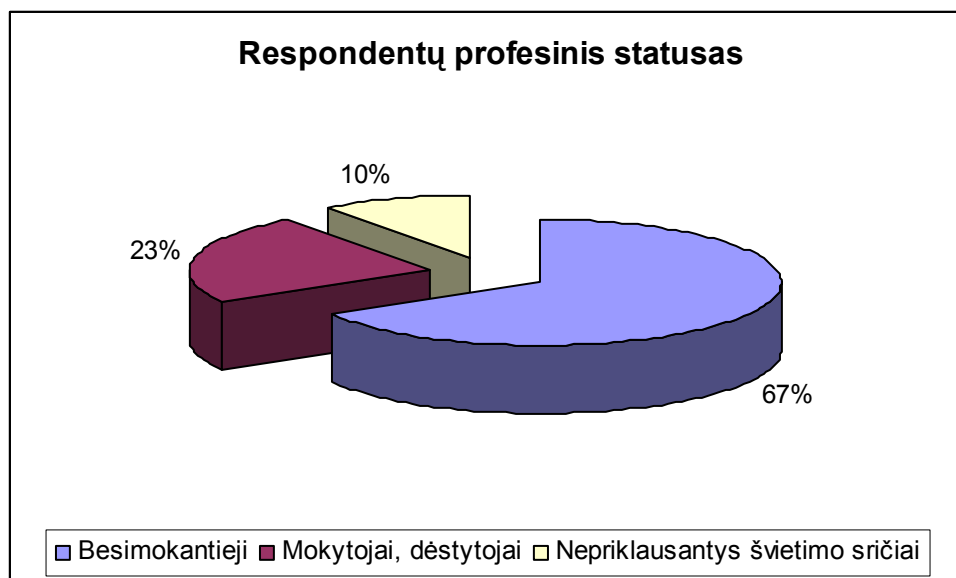
1. Baniulis K. T., Reklaitis V., Stuopys E. Interactive multimedia authoring tool for web-based testing // Proceedings of ED-MEDIA 2001: World conference on Educational Multimedia, Hypermedia and Telecommunications. Norfolk: AACE, 2001, p. 84-85.
2. Ambrazevičius E., Šakys V., Jasiukevičius A. Universitetinių e. studijų sistemų kūrimo principai ir problemos // Informacijos mokslai: mokslo darbai / Vilniaus universitetas, 2002, T, 22, p. 19-29.
3. Telešius E., Otas A. ECDL programos startas Lietuvoje: pirmieji žingsniai ir nuotolinio mokymo perspektyvos // Atvirojo ir distancinio mokymosi vaidmuo informacinėje visuomenėje. Kaunas: Technologija, 1999, p.155-160.
4. Telešius E. Skills of Information Technologies - to everyone ECDL Program in Lithuania - Strategic Initiative of Computer Society // Informacijos visuomenė 99: Tarpt. konf. praneš. Vilnius: Infobalt Lietuva, 1999. p.178-180.
5. Grigas D., Tamulynas B. Programinių modulių kokybės vertinimas kompiuterizuotoje žinių testavimo sistemoje // Pažangios informacinės technologijos ir jų taikymas studijų procese: tarptautinės konferencijos pranešimų medžiaga / Alytus: Alytaus kolegija, 2002, p. 113-118.
6. Paulauskas M.P. Skaitmeninė Lietuva 2001.V.: Vilniaus universiteto I-kla, 2001, p. 68-77
7. Kasanavičius E., Kasanavičius V. Informacijos apdorojimo technologijos. K.:Kauno technologijos universiteto I-kla, 2002.
8. Lietuvos ECDL testavimo centro tinklalapis. Prieiga per Internetą:
<http://www.ecdl.lt>. [žiūrėta 2004-03-22]
9. MEDIAPlus įrankio tinklalapis. Prieiga per Internetą:
<http://www.testoffice.com/English/Training/signin.asp> [žiūrėta 2003-10-27]
10. Jakarta Tomcat projekto tinklalapis. Prieiga per Internetą:
<http://jakarta.apache.org/tomcat> [žiūrėta 2003-11-25]
11. Brainbench žinių vertinimo sistemos tinklalapis. Prieiga per internetą:
<http://www.brainbench.com> [žiūrėta 2004-03-26]
12. WBTwise sistemos tinklalapis. Prieiga per internetą: <http://www.wbtwise.com> [žiūrėta 2004-01-15]
13. Questionmark sistemos tinklalapis. Prieiga per internetą: <http://www.questionmark.com> [žiūrėta 2004-04-19]
14. „The Teacher Training Agency“ sistemos tinklalapis. Prieiga per internetą:
<http://www.canteach.gov.uk/support/skillstests/ict/index.htm> [žiūrėta 2003-09-24]
15. The J2EE Tutorial. Prieiga per Internetą: <http://java.sun.com/j2ee/tutorial> [žiūrėta 2004-01-10]
16. IBM DBVS tinklalapis. Prieiga per internetą: <http://www.IBM.com> [žiūrėta 2004-01-22]

8. Terminų ir santrumpų žodynas

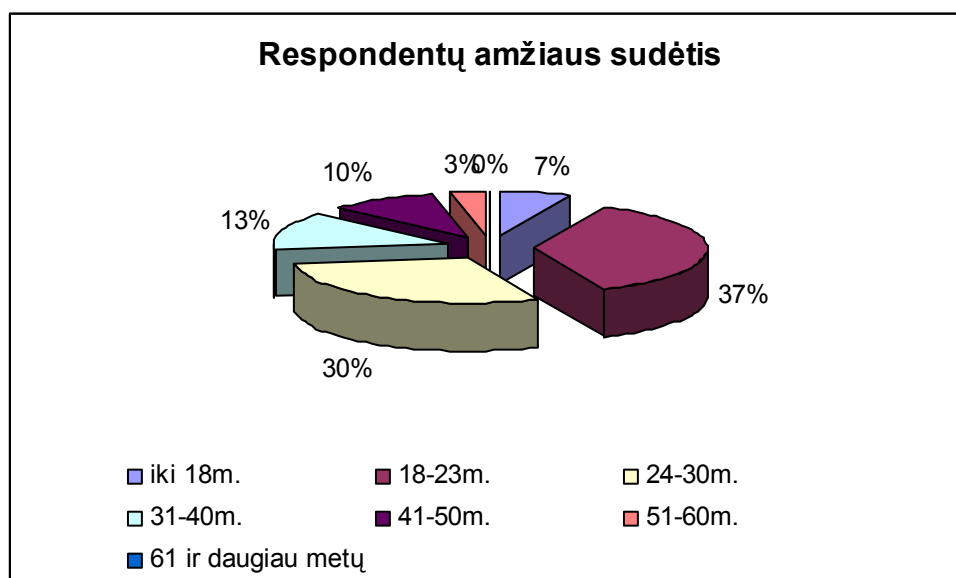
| Terminas | Apibrėžimas |
|----------------------|--|
| DBVS | duomenų bazių valdymo sistema |
| Java | objektinė programavimo kalba |
| JDBC | sujungimo su duomenų bazėmis Java programavimo kalba technologija |
| JSP | Java Server Pages – dinaminių puslapių integruojantys Java kodą |
| Servletų konteineris | interneto serveris apdorojantis dinaminis JSP puslapius bei Java komponentus, naudojamus internetinei aplikacijai. |
| DB2 | IBM sukurta duomenų bazių valdymo sistema |
| Tomcat | “Jakarta” projekto dalyvių sukurtas servletų konteineris |
| PĮ | Programinė įranga |
| PS | Programų sistema |

9. Priedai

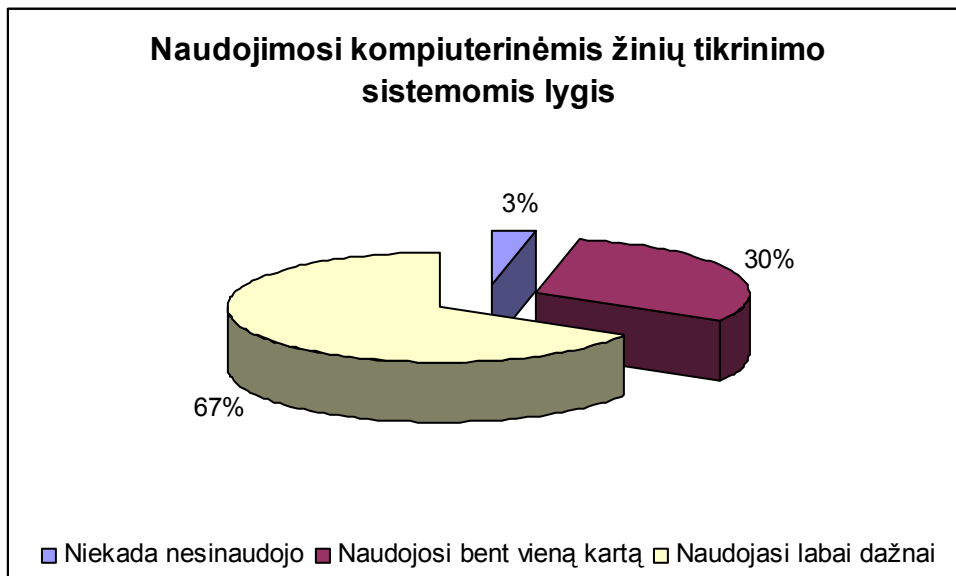
1 Priedas. Apklaustųjų profesinis statusas



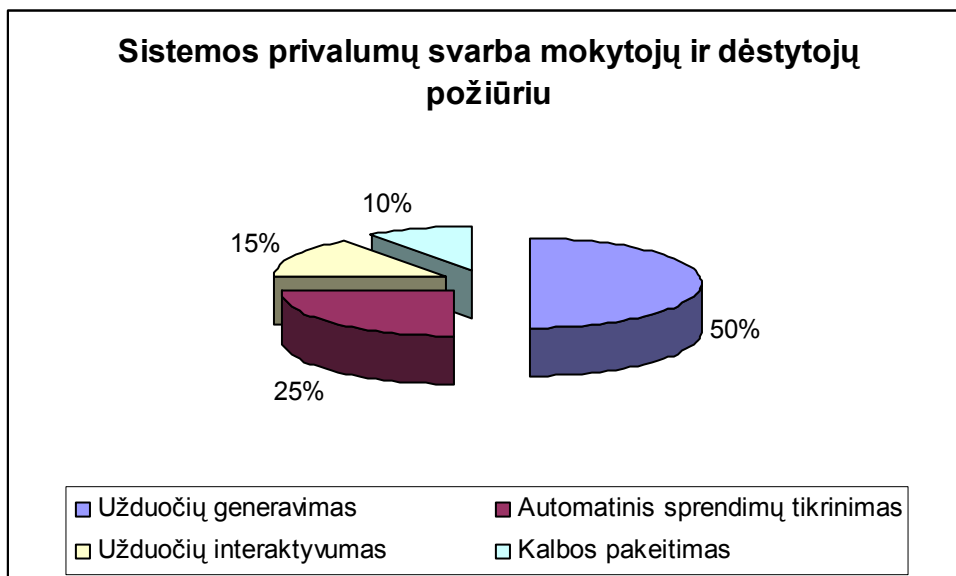
2 Priedas. Apklaustųjų amžiaus sudėtis



3 Priedas. Apklaustųjų naudojimas žinių tikrinimo sistemomis



4 Priedas. Sistemos savybių svarba dėstytojų požiūriu



5 Priedas. **Sistemos savybių svarba besimokančiųjų požiūriu**

