

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Laimis Šeškevičius

**Duomenų perdavimas tarp ActionScript ir MySQL
formuojant animacinius objektus**

Magistro darbas

Darbo vadovas
doc. dr. E. Karčiauskas

KAUNAS, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Laimis Šeškevičius

**Duomenų perdavimas tarp ActionScript ir MySQL
formuojant animacinius objektus**

Magistro darbas

Recenzentas

doc. dr. E. Mačikėnas

2009-05-25

Vadovas

doc. dr. E. Karčiauskas

2009-05-25

Atliko

IFM-1/5 gr. Stud.

Laimis Šeškevičius

2009-05-25

KAUNAS, 2009

Turinys

LENTELIŲ SĄRAŠAS	5
PAVEIKSLŲ SĄRAŠAS	5
SUMMARY	6
SANTRAUKA	6
1. ANALITINĖ DALIS	7
1.1. Sistemos paskirtis	7
1.2. Naudotos technologijos	9
1.2.1. XHTML	9
1.2.2. JavaScript	10
1.2.3. CSS	10
1.2.4. MySQL	11
1.2.5. PHP	11
1.2.6. ActionScript	12
1.3. Problemos	12
1.4. Duomenų maršrutai	13
1.5. GET ir POST metodų palyginimas	14
1.5.1. GET metodas	14
1.5.2. POST metodas	15
1.5.3. Metodų palyginimas	15
1.6. XML ir LoadVars klasių palyginimas	16
1.6.1. XML	16
1.6.2. XML duomenų įkėlimas	17
1.6.3. XML duomenų išsiuntimas	17
1.6.4. LoadVars klasė	18
1.6.5. LoadVars duomenų įkėlimas	18
1.6.6. LoadVars duomenų siuntimas	19
1.7. Derinių atvaizdavimas	19
1.8. Apibendrinimas	21
2. PROJEKTINĖ DALIS	22
2.1. Panaudojimo atvejai	24
2.2. Panaudojimo atvejų sąrašas	24
2.3. Aktorių funkcijos	27
2.3.1. Sportininkas	27
2.3.2. Studentas	27
2.3.3. Administratorius	28
2.4. Loginis vaizdas	29
2.5. Funkciniai reikalavimai	30
2.6. Nefunkciniai reikalavimai	31
2.6.1. Reikalavimai sistemos išvaizdai	31

2.6.2.	Reikalavimai panaudojamumui	31
2.6.3.	Reikalavimai vykdymo charakteristikoms.....	31
2.6.4.	Reikalavimai veikimo sąlygoms	31
2.6.5.	Reikalavimai sistemos priežiūrai	31
2.6.6.	Reikalavimai saugumui.....	31
2.6.7.	Kultūriniai/politiniai reikalavimai	32
2.6.8.	Teisiniai reikalavimai	32
2.7.	Veiklos diagrama	32
2.8.	Sistemos išdėstymo vaizdas	33
2.9.	Serverio reikalavimai	34
2.10.	Vartotojo kompiuterio reikalavimai.....	34
3.	EKSPERIMENTINĖ DALIS	35
3.1.	Eksperimentų tikslas	35
3.2.	Duomenų perdavimo eksperimentas su nejudančiais aktorais.....	35
3.3.	Duomenų perdavimo eksperimentas su judančiais aktorais	36
3.4.	Animuojamų objektų greitaveikos eksperimentas	37
3.5.	Eksperimentų apibendrinimas	37
4.	IŠVADOS	38
5.	LITERATŪROS SĄRAŠAS	39
6.	TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	41
7.	PRIEDAI.....	42

LENTELIŲ SĄRAŠAS

<i>1.5.3.1. lentelė. GET ir POST metodų palyginimas</i>	15
<i>1.6.1.1. lentelė. XML failo pavyzdys</i>	17
<i>2.1. lentelė. DB Esybių paaiškinimai</i>	23
<i>2.9.1. lentelė. Reikalavimai serveriui</i>	34
<i>2.10.1. lentelė. Reikalavimai klientui</i>	34

PAVEIKSLŲ SĄRAŠAS

<i>1.4.1. paveikslas. Informacijos saugojimas</i>	13
<i>1.4.2. paveikslas. Informacijos atvaizdavimas</i>	13
<i>1.7.1. paveikslas. Derinio atvaizdavimo schema</i>	20
<i>2.1. paveikslas. Duomenų bazės schema</i>	22
<i>2.1.1. paveikslas. Informacinės sistemos panaudojimo atvejai</i>	24
<i>2.4.1. paveikslas. Sistemos išskaidymas į paketus</i>	29
<i>2.7.1. paveikslas. Veiklos diagrama</i>	32
<i>2.8.1. paveikslas. Išdėstymo diagrama</i>	33
<i>3.2.1. paveikslas. Duomenų kiekio nuo aktorių skaičiaus priklausomybė</i>	35
<i>3.3.1. paveikslas. Duomenų kiekio nuo kadru skaičiaus priklausomybė</i>	36
<i>3.4.1. paveikslas. Aktorių kiekio animacijos vertinimas</i>	37
<i>7.1. paveikslas. Derinių kūrimo objekto pavyzdys</i>	42

SUMARRY

Data transfer between ActionScript and MySQL forming animated objects

In master work was created animated games drills information system. Also where was analysis of data transfer possibilities between ActionScript (media programming language) and MySQL (data base). Where in MySQL data base is all required information for animation: stripe parameters, coordinates. When this information is given to ActionScript object it creates animated drills. With implemented information system it is possible to create most popular sport games drills, later the drills can be reviewed, commented, changed.

Implemented information system was completed successfully, where was done the most important components testing. When few experiments was made, results showed, that data transfer amount with many stripes is small, because where is used special data transfer structure. ActionScript object creates drills with many sprites very fast. When the experiment with many sprites was done, results show, that it is possible to animate up to 1000 sprites at the same time, what this information system makes very attractive not only for making sport drills, but it also can be used for other proposes, where many sprites need to be animated.

SANTRAUKA

Magistrinio darbo metu buvo realizuojama animuotų žaidimų derinių informacinė sistema. Taipogi buvo analizuojamos duomenų perdavimo galimybės tarp ActionScript (animacijos programavimo kalbos) ir MySQL (duomenų bazės). Kur MySQL duomenų bazėje yra saugoma visa reikalinga informacija animaciniai sudaryti: aktorių parametrai, koordinatės. Informaciją perdavus ActionScript objektui, objektas sukuria animuotus derinius. Realizuotos informacinės sistemos pagalba galima kurti populiariausių komandinių sporto šakų derinius, vėliau juos galima peržiūrėti, komentuoti, redaguoti.

Realizuota informacinė sistema buvo sėkmingai sukurta, atliktas svarbiausių komponentų testavimas. Atlikus keletą eksperimentų, rezultatai parodė, kad duomenų perdavimo kiekis esant dideliame kiekiui aktorių yra mažas, nes yra naudojamas speciali duomenų perdavimo struktūra. ActionScript objektas derinius atkuria labai greitai net esant dideliame aktorių kiekiui. Atlikus eksperimentus su daugybe aktorių, rezultatai parodė, kad galima sėkmingai animuoti iki 1000 aktorių vienu metu, kas informacinę sistemą daro patrauklia ne tik kuriant sportinius derinius, bet galima pritaikyti ir kitose srityse, kur reikia animuoti daugybę aktorių.

1. ANALITINĖ DALIS

1.1. Sistemos paskirtis

Šiuo metu tokios tematikos programinės įrangos pasirinkimas nėra didelis. Toks projektas būtų labai geras pagalbininkas įvairių sporto šakų treneriams (krepšinis, futbolas, rankinis, tenisas, ledo ritulys ir t.t.), žaidėjams, sporto mėgėjams. Animuotų žaidimų derinių informacinės sistemos pagalba būtų galima animuoti daugybę įvairių sporto šakų derinių, kurie yra sukaupti daugybėje sporto derinių knygų.

Sporto šakų treneriai šio įrankio pagalba galėtų savo auklėtiniams lengviau sudarinėti įvairiausių derinius, kur deriniai būtų ne nuotraukos (kaip yra vaizduojama knygoje), o realiu laiku judantys žaidėjai. Šią sistemą patalpinus interneto svetainėje būtų galima diskutuoti apie konkrečius derinius su kitais treneriais. Sukūrus nemažą derinių duomenų bazę, būtų galima pagal pasirinktus kriterijus greitai atlikti norimų derinių paiešką, ko pasėkoje būtų taupomas laikas.

Trenerių auklėtiniams (žaidėjams) tokia sistema praverstų, kai jiems reikėtų išmokyti konkrečius derinius. Matant realiu laiku judančius derinius būtų lengviau išmokstami deriniai. Ko pasėkoje žaidėjas galėtų lengviau įsijungti į komandą. Nauji komandų žaidėjai derinius galėtų iš anksto analizuoti būdami, bet kurioje pasaulio vietoje, kur yra interneto prieiga. Tokiems žaidėjams norint turėti prieigą prie konkrečių derinių, turės būti pasakyti prisijungimo duomenys prie informacinės sistemos.

Taigi šios sistemos pagalba būtų galima patogiai atvaizduoti, judančių žaidėjų derinių atvaizdavimas. Derinių analizė pasidarytų daug paprastesnė ir aiškesnė. Derinių informacinę sistemą būtų galima pasiekti iš, bet kurios pasaulio vietos, kur yra prieiga prie interneto.

Atlikus panašių produktų paiešką (internete) Lietuvoje, rezultatai parodė, kad panašių produktų nėra. Pasaulyje tokio tipo produktų pavyko surasti. Bet jie visi buvo specializuoti į konkrečią sporto šaką: futbolas, krepšinis. O tokio bendro stuburo, kurio pagalba būtų galima animuoti įvairių sporto šakų derinius surasti nepavyko.

Tokio pobūdžio informacinė sistema nepamaišytų krepšinio, futbolo, rankinio sporto klubams. Šią sistemą būtų stengiamasi labiau orientuoti į komandinio sporto šakas, kadangi pas jas yra didesni biudžetai. Įvertinant tai, kad Lietuvoje tokio pobūdžio informacinių sistemų nepavyko surasti galima teigti, kad tokia sistema turėtų paklausa.

Grįžtant prie derinių knygų, dabartiniai šių knygų kūrėjai galėtų prie knygos pridėti kompaktinį diską, kurio pagalba būtų galima peržiūrėti, analizuoti knygoje aprašomus derinius.

Tam, kad knygos autorius galėtų tai atlikti jis turėtų įsigyti šią sistemą. Knygų autoriams skirta sistema veiktų tikrai pas galinius vartotojus („end user“), tokia sistema skirtųsi nuo internetinės sistemos. Ši sistema būtų stipriai apkarpyta tam, kad nebūtų tinkama įvairių sporto šakų treneriams. Sistemos įrankių pagalba būtų galima sukurtus derinius, kuriuos būtų galima lengvai patalpinti kompaktiniuose diskuose, kurie būtų knygų priedai.

Internetui skirta sistema būtų mokama. Norint gauti galimybę naudotis sistema reikėtų sumokėti abonentinį mokestį, už kurį vartotojui būtų suteikiamas prisijungimo vartotojas ir slaptažodis. Mokesčio dydis priklausytų nuo pageidaujamų patalpinti derinių kiekio ir vartotojų kiekio, kuris galės prieiti prie konkrečios duomenų bazės. Norint turėti didesnę derinių duomenų bazę reikėtų daugiau ir mokėti.

Taigi pavyzdžiui jeigu viena krepšinio komanda norėtų gauti savo visos komandos žaidėjams priėjimą prie jų komandos derinių duomenų bazės. Ji turėtų nusipirkti vartotojų priėjimą prie jų duomenų bazės. Kad nebūtų sukčiavimų (turint vieną prisijungimą visi komandos žaidėjai gali prisijungti per tą patį vartotoją) vienu metu konkretus vartotojas galėtų tikrai vieną kartą prisijungti prie informacinės sistemos. Kitaip sakant tuo pačiu metu nebūtų galimybės dviem (ar daugiau) asmenims esantiems prie skirtingų kompiuterių prisijungti prie sistemos. Vienos komandos deriniai būtų prieinami tikrai konkrečiai komandai.

Norint, kad komandoms reikėtų mažiau mokėti už jų žaidėjų prisijungimus, jie galėtų tam tikrus savo derinius paskelbti laisvai prieinamais. Tokiu atveju iš daugybės komandų tokių laisvai prieinamų derinių susidarytų nemažas kiekis derinių. Šie deriniai būtų prieinami sporto šakų mėgėjams, entuziastams. Kurie sumokėję nedidelį abonento mokestį galėtų peržiūrėti šiuos laisvai prieinamus derinius. Kuo konkreti komanda daugiau derinių paskelbtų laisvai prieinamais, tuo jai tektų mažiau mokėti už abonentus.

Laikui bėgant sistemą būtų galima papildyti papildomomis funkcijomis už kurias jei komandos pageidautų reikėtų papildomai sumokėti.

Taigi didžiausio pelno būtų siekiama iš krepšinio, futbolo, rankinio ekipų, kurios norėdamos naudotis sistema turėtų mokėti abonentinius mokesčius. Kitas pasipelnymo būdas būtų tokių sistemų pardavimas galiniams vartotojams, tokios sistemos būtų daug paprastesnės ir turėtų mažiau galimybių nuo tos, kuri būtų patalpinta internete.

1.2. Naudotos technologijos

Kuriant informacinę sistemą buvo panaudotos šios technologijos:

- XHTML 1.0
- JavaScript
- CSS
- MySQL 5.0
- PHP 5.2.9
- ActionScript 3.0

1.2.1. XHTML

XHTML - (eXtensible HyperText Markup Language – praplėsta hiperteksto žymėjimo kalba) yra šeima dabartinių ir būsimų dokumentų tipų, modulių, kurie praplečia HTML 4. XHTML dokumentų tipų šeima yra paremta XML, kuri yra sukurta dirbti sąsajoje su XML.

XHTML 1.0 yra pirmasis dokumento tipas XHTML šeimoje. Jis yra perdarytas HTML 4 dokumentas, kuris tarsi yra XML parašyta programa. XHTML ketinta naudoti kaip kalbą atitinkančią tiek XML ir HTML. XHTML turi šiuos privalumus:

- XHTML dokumentai atitinka XML. Jie yra lengvai peržiūrimi, redaguojami ir validuojami standartiniais XML įrankiais.
- XHTML dokumentai gali būti parašyti taip pat (HTML) arba geriau negu jie buvo rašomi.
- XHTML dokumentai gali naudoti programas, kurie remiasi HTML DOM (Document Object Model) arba XML DOM.
- Plėtojantis XHTML šeimai, dokumentus atitinkančius XHTML 1.0 ateityje bus lengviau pritaikyti įvairioms XHTML aplinkoms [1].

XHTML šeima yra sekantis žingsnis interneto evoliucijoje. Perinant prie XHTML šiandien, kūrėjai patenka į XML pasaulį, su visais jį lydinčiais privalumais, tuo pat metu palaikant egzistuojančius standartus bei suderinamumą ateičiai [1].

1.2.2. JavaScript

Populiari skriptų kalba, kuri yra palaikoma daugybės interneto naršyklių ir kitų internetinių įrankių. Ji suteikia interaktyvumo funkcijų HTML puslapiams, be JavaScript jie būna statiniai, kol HTML yra atvaizdavimo, o ne programavimo kalba. JavaScript yra lengviau naudojamas negu Java, bet ne toks galingas ir daugiausiai, dažniausiai yra

Naudojamas internetinių puslapių elementuose. Pas klientą JavaScript yra pateikiamas kaip pirminis kodas įterptas į HTML puslapį. Serveryje jisai yra sukompiliuojamas, panašiai kaip Java programose [2].

JavaScript išsivystė iš Netscape LiveScript kalbos. Pirmą kartą buvo išleistas su Navigator 2.0, kuris buvo labiau suderinamas su Java. JavaScript neturi programinio ryšio su Java, bet gali būti naudojami sąjungoje vienas su kitu. Pavyzdžiui, Javascript skriptas gali būti panaudotas atvaizduojant duomenų lauką bei patikrinti įvestį, kur Java programa vėliau apdoroja informaciją.

1.2.3. CSS

CSS - (Cascading Style Sheet) Stiliaus šablonai yra naudojami sudarant internetinių puslapių išdėstymą. Jie gali būti naudojami apibrėžiant teksto stilių, lentelių dydžius bei kituose internetinių puslapių elementuose, kurie anksčiau galėdavo būti apibrėžiami tik HTML puslapiuose.

CSS internetinių puslapių kūrėjams padeda sukurti pastovų atvaizdavimą keliuose interneto svetainės puslapiuose. Vietoj to, kad nurodytume stilių kiekvienai lentelei ar kiekvienam bloko tekstui HTML puslapyje, dažnai naudojami stiliai turi būti aprašyti tik kartą CSS dokumente. Kai stilius būna aprašytas stiliaus šablone, jis gali būti naudojamas kiekviename puslapyje, kuris turi nuorodą į CSS failą. CSS pagalba galima lengvai pakeisti stilius keliuose interneto puslapiuose iš karto. Pavyzdžiui, puslapio kūrėjas nori padidinti standartinį teksto dydį nuo 10pt iki 12pt penkiasdešimtyje svetainės puslapių, jeigu visi puslapiai turi nuorodą į konkretų stiliaus šabloną, teksto dydis gali būti pakeistas, pakeičiant teksto dydį stiliaus šablonų faile ir visi puslapiai pradės rodyti didesnę tekstą [3].

Nors CSS yra patogus kuriant tekstinius stilius, jisai padeda sukurti stiliaus šablonus kituose interneto puslapio elementuose taip pat. Pavyzdžiui, CSS gali būti naudojamas nurodant lentelių spalvą, tarpus tarp ląstelių, tarpus aplink nuotraukas ar kitus objektus. CSS suteikia puslapių kūrėjams daugiau galimybių negu interneto puslapiai gali atrodyti HTML. Dėl šių priežasčių didžioji dauguma šių dienų interneto puslapių yra apjungti su stiliaus šablonais.

1.2.4. MySQL

MySQL – yra atvirojo kodo reliacinė duomenų bazių valdymo sistema. Jos pagrindas yra SQL (Structure Query Language – struktūrinė užklausų kalba), kuri yra naudojama įtraukiant, šalinant ir modifikuojant informaciją duomenų bazėje. Standartinės SQL komandos, tokios kaip SELECT, DELETE, INSERT, UPDATE gali būti naudojamos MySQL duomenų bazėje [4], [5].

MySQL gali būti naudojama daugybėje programų, bet dažniausiai ši duomenų bazė aptinkama interneto serveriuose [6]. Interneto svetainė, kuri naudoja MySQL gali įtraukti internetinius puslapius, kurie gauna informaciją iš duomenų bazės [7]. Tokie puslapiai neretai būna vadinami dinaminiais, kas reiškia, kad puslapio turinys yra generuojamas iš duomenų bazės kiekvieną kartą puslapiui sudarant.

Dauguma svetainių su duomenų bazėmis, kurios naudoja MySQL taipogi naudoja internetinę dinaminę interpretuojamą programavimo kalbą (PHP) norint gauti informaciją iš duomenų bazės. MySQL komandos suvienytos su PHP programinių kodu, taip suteikiant galimybę daliai arba visam internetiniam puslapiui būti sugeneruotam iš duomenų bazės gautos informacijos [8]. Kadangi MySQL ir PHP yra atvirojo kodo (parsisiuntimas nieko nekainuoja), PHP/MySQL kombinacija tapo labai populiariu pasirinkimu dinaminėse svetainėse [9].

1.2.5. PHP

PHP – yra į HTML įdiegiama internetinė skriptų kalba. Tai reiškia, kad PHP programinis kodas gali būti įkeltas į HTML internetiniuose puslapiuose. Kada PHP puslapis yra aktyvuojamas, PHP kodas yra perskaitomas serverio, kuriame jisai egzistuoja. PHP funkcijų išvestis puslapiuose dažniausiai grąžinamas kaip HTML kodas, kuris gali būti perskaitymas interneto naršyklių. Dėl to, kad PHP kodas yra transformuojamas į HTML prieš puslapiui užsikraunant, vartotojai negali pamatyti puslapyje esančio PHP kodo [10],[11]. Tai PHP puslapius daro pakankamai saugius turint priėjimą prie duomenų bazių ar kitokios svarbios informacijos.

Didžioji dauguma PHP kalbos sintaksės yra pasiskolinta iš kitų kalbų tokių kaip C, Java ir Perl. Nors PHP turi daugybę unikalių savybių ir specialių funkcijų taip pat [12]. Šios kalbos tikslas yra suteikti galimybę internetinių puslapių kūrėjams rašyti dinamiškai generuojamus puslapius greitai ir paprastai. PHP yra labai patogus kuriant interneto svetaines su duomenų bazėmis.

1.2.6. ActionScript

ActionScript yra skriptų kalba sukurta pagal ECMAScript. Actionscript yra dažniausiai yra naudojamas interneto svetainių kūrime ir programinė įranga naudojanti Adobe Flash grotuvo platformą (su SWF failais įklijuotais interneto svetainėse). Sukurtas kompanijos Macromedia, šiuo metu priklauso Adobe kompanijai, kurią įsigijo Macromedia kompaniją 2005 metais. Nuo pat pradžių ActionScript buvo sukurtas valdyti dvimatę vektorinę animaciją sukurta Adobe Flash [13], [14]. Vėlesnėse versijose pridėtas funkcionalumas leido sukurti internetinius žaidimus ir turtingas internetines programas su srautine media: vaizdo ir garso.

1.3. Problemos

- ActionScript savyje neturi reikalingų funkcijų, kad pats savaime galėtų prisijungti prie MySQL duomenų bazių [15]. Norint perduoti duomenis iš ActionScript objekto į MySQL duomenų bazę reikalingas tarpininkas. Kadangi su ActionScript objektu yra galimybė perduoti duomenis į PHP, kaip tarpininkas buvo pasirinkta ši kalba [16].
- Duomenis iš ActionScript į PHP galima perduoti dviem būdais GET arba POST metodais (jie yra analogiški GET ir POST metodams naudojamiems HTML kalboje). Kuriant informacinę sistemą reikėjo padaryti šių dviejų metodų palyginimą ir pasirinkti tinkamesnį.
- Duomenis iš ActionScript į PHP galima perduoti dviem būdais *LoadVars* klasės pagalba arba *XML* klasės pagalba. Perduodant duomenis reikėjo pasirinkti ir naudoti vieną iš šių klasių.
- Duomenų perdavimas iš ActionScript į MySQL duomenų bazę ir duomenų paėmimas iš duomenų bazės į ActionScript objektą atliekamas skirtingais būdais.
- Duomenų paėmimas į ActionScript objektą veikia panašiu principu kaip AJAX [17]. ActionScript objektas kreipiasi į tam skirtą failą, iš kurio gauna visą informaciją (animacinių objektų koordinatės, dydžiai, spalvos ir t.t.), kurios pagalba yra atkuriami animaciniai objektai. Su sunkumais yra susiduriama, kai kiekvieną kartą kreipiantis į nutolusį failą informacija jame skiriasi [18], [19].

1.4. Duomenų maršrutai

Derinių saugojimo procedūra:

- Derinio sukūrimas.
- ActionScript iš visų duomenų sudaro masyvą.
- Masyvas perduodamas PHP.
- PHP gautą duomenų masyvą sukarpo bei išsaugo MySQL duomenų bazėje atitinkamose lentelėse.



1.4.1. paveikslas. Informacijos saugojimas

Derinių atvaizdavimo procedūra:

- Visų derinio duomenų surinkimas pagal derinio identifikacinį numerį.
- Duomenis perdavus PHP jisai iš jų padaro nustatytos struktūros masyvą.
- Masyvas perduodamas ActionScript objektui.
- ActionScript objektas masyvą suskaldo, iš gautų duomenų būna sukuriami aktoriai, rodyklės, derinio fonas ir t.t.



1.4.2. paveikslas. Informacijos atvaizdavimas

1.5. GET ir POST metodų palyginimas

HTTP yra interneto protokolas. Tai yra tai, kad perduoda duomenis tarp kliento į serverį ir atgal. HTTP specifikacija apibrėžia keletą HTTP metodų perduodamiems skirtingiems duomenų tipams. HTTP metodai: GET ir POST yra pagrindiniai metodai naudojami internete šiandien.

1.5.1. GET metodas

Šiuo metodu duomenys yra perduodami URL adrese. Kintamieji URL būna URL adreso gale. Kintamieji nuo internetinio adreso būna atskirti klaustuko (?) simboliu, po skyriklio eina kintamojo vardas tada lygybės (=) ženklas, po kurio būna kintamojo reikšmė. Jeigu būna daugiau nei vienas kintamasis jie būna atskiriami simboliu (&). Pavyzdys:

<http://www.adresas.lt?alfa=1&beta=raide&gama=0>

Aukščiau pateiktame pavyzdyje GET metodu būtų perduodami trys kintamieji alfa, beta ir gama. Alfa reikšmė būtų 1, beta reikšmė ‚raidė‘, o gama reikšmė 0.

Šiuo metodu negalima perduoti tikrai ribotą kiekį duomenų. Oficialiai URL adresas negali būti ilgesnis negu 256 simboliai, bet šis limitas būna dažnai viršijamas. Egzistuoja daugybė serverių, kurie turi savo nustatytus maksimalius URL adresų ilgius, taip didinamas serverio saugumas. Populiariausia šiuo metų esanti interneto naršyklė (Internet Explorer) oficialiai palaiko iki 2083 simbolių. Kitos naršyklės tokios kaip: Opera, Firefox, Safari palaiko keliais tūkstančiais daugiau simbolių negu Internet Explorer, bet žinant, kad šią naršyklę naudoja daugiausiai vartotojų, ilgesnius URL adresus sudarinėti būtų nenaudinga ir nepraktiška.

Duomenys perduodami GET metodu turi apribojimus. Galima perduoti tikrai ASCII lentelės simbolius. Šis apribojimas nesuteikia galimybės šiuo metodu perduoti specialių simbolių naudojamų kitose kalbose (lietuvių, rusų, čekų ir t.t.). Tokius specialius simbolius galima perduoti kintamųjų reikšmes užkoduojant vėliau iššifruojant, bet toks būdas nėra labai patogus, o užkoduoti simboliai turi būti koduojami keliais simboliais todėl naudojant specialius simbolius, URL adresas gali pasidaryti labai ilgas. Perduodami duomenys gali būti tikrai tekstinio formato. GET metodu negalima perduoti binarinės (binary) duomenų: vaizdo failų, garso failų ir t.t.

Šiuo metodu perduodami duomenys nėra saugūs, nes juos visi nesunkiai pastebimi naršyklės URL adrese. Šiuo metodu nėra patartina neperdavinti slaptažodžiu, prisijungimo vardų ar kitų svarbesnių duomenų. Taipogi šiuo metodu galima įvykdyti SQL injekcijas, kurių metu galima gauti prieigą prie duomenų bazės duomenų [20], prieš SQL injekcijas turi būti naudojamos specialios apsaugos funkcijos. Šiuo metodu perduodamų kintamųjų reikšmes patyrę vartotojai gali

nesunkiai redaguoti, ištrinti vieną ar kelis kintamuosius nesunkai pakeičiant GET metodu perduodamą informaciją [21].

Šis metodas yra labai patogus atliekant testus internetinių puslapių kūrėjams. Testavimą palengvina tai, kad visada yra matomi GET metodu perduodami kintamieji. Šis metodas yra labai patogus, kai reikia perduoti trumpą informaciją, kuri dažniausiai neturi ryšio su duomenų baze.

1.5.2. POST metodas

Šis duomenų perdavimo metodas yra daug saugesnis negu GET metodas. Jo nebūna URL adrese. Šis metodas yra naudojamas perduodant svarbią informaciją, prisijungimo vartotojo vardus, slaptažodžius, informaciją paimtą iš duomenų bazės. Kitaip nei GET metodo atveju vartotojas negali redaguoti perduodamos informacijos. Yra galimybė perduodamą informaciją paslėpti nuo vartotojo (hide), skirtumas nuo GET metodo yra tas, kad šiuo metodu perduodamos informacijos vartotojas negali modifikuoti.

Šiuo metodu galima perduoti tiek tekstinę, tiek binarinę informaciją. Šiuo metodu galima perduoti vaizdinę informaciją, garsinę ar tekstinius failus. Nėra apribojimo tarp perduodamų simbolių, galima perduoti, bet kokius simbolius be jokių specialių lentelių apribojimų. Naudojant standartinius serverio Apache ir PHP nustatymus galima perduoti iki 8mb duomenų. Maksimalų perduodamų duomenų kiekį galima mažinti arba didinti. Šis metodas yra geriausias, kai reikia perduoti didelį duomenų kiekį.

Interneto puslapių kūrėjams metodas yra sunkiau testuojamas negu GET metodas. Metodas tinkamas atliekant operacijas su duomenų bazėje esančia informacija.

1.5.3. Metodų palyginimas

1.5.3.1. lentelė. GET ir POST metodų palyginimas

Kriterijus	GET metodas	POST metodas
Saugumas		+
Didelių duomenų perdavimas		+
Lietuviškų simbolių perdavimas		+
Paprastumas naudoti	+	
Binarinės informacijos perdavimas		+
Testavimas	+	

Pateiktoje lentelėje metodai buvo palyginti pagal tam tikrus kriterijus. Pliusas reiškia, kad konkrečiu aspektu metodas yra pranašesnis:

- **Saugumas:** POST metodas yra saugesnis už GET metodą.
- **Didelių duomenų perdavimas:** POST metodu galima perduoti daug kartų didesnę duomenų kiekį negu GET metodu.
- **Lietuviškų simbolių perdavimas:** POST metodu tai nesukelia jokių papildomų problemų, GET metodo atveju reikalingas simbolių kodavimas.
- **Paprastumas naudoti:** GET metodu perduodami duomenys lengviau pastebimi, visada matosi perduodami kintamieji, POST metodu kintamieji gali būti slepiami.
- **Binarinės informacijos perdavimas:** GET metodu galima perduoti tik tekstinę informaciją, POST metodu tekstinę, vaizdinę, garsinę ir kitokio pobūdžio informaciją, failus.
- **Testavimas:** Kadangi GET metodu perduodami kintamieji yra lengviau pastebimi jisai yra lengviau testuojamas, GET metodo atveju galima lengviau sugeneruoti automatinius testus, negu POST metodo atveju.

Įvertinus šiuos esminius metodų aspektus informacinėje sistemoje buvo nuspręsta pasirinkti naudoti POST metodą.

Svarbiausias kriterijus - didelių duomenų perdavimo galimybė. Šiuo aspektu GET metodu būtų galima perduoti labai mažą informacijos kiekį, kurio neužtektų norint atvaizduoti didesnę animuojamų elementų kiekį. Taipogi turėjo būti galimybė perduoti lietuviškus rašmenis, POST metodas tam yra daug parankesnis, pasiekiant galimybę ateityje sistemoje naudoti kitų kalbų specialius simbolius.

1.6. XML ir LoadVars klasių palyginimas

1.6.1. XML

XML (Extensible Markup Language) yra lankstus būdas sukurti bendrus informacijos formatus ir dalintis formatu ir informacija internete, intranete ar kitur. Pavyzdžiui, kompiuterių gamintojai gali susitarti dėl bendro standarto ir bendro būdo aprašant informaciją kompiuterio sudedamąsias dalis (procesorius, greitis, atmintis, didis ir t.t.) ir tada aprašyti produktų informaciją su XML formatu.

Toks standartinis būdu aprašytus duomenis vartotojas galės išsiųsti programą visiems tokių prekių pardavėjams kaip siūlymą rinkti duomenis vėliau juos naudojant vienodame formate. XML gali būti naudojamas vienoje kompanijoje arba keliose kompanijose, kurios pageidauja dalintis informacija pastovu būdu.

1.6.1.1. lentelė. XML failo pavyzdys

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<žinutė>
    <kam>Tomui</kam>
    <nuo>Mariaus</nuo>
    <antraštė>Priminimas</antraštė>
    <turinys>Kada gražinsi skola!</turinys>
</žinutė>
```

1.6.2. XML duomenų įkėlimas

XML dokumentams nesantiems Flash filme, pirma reikia įkelti tekstą į filmą. Šį darbą atlieka load() metodas. Metodas load() įkelia XML dokumentą iš išorinio šaltinio į Flash filmą, bei sukuria XML medį sukurtam XML objektui.

Metodui load() yra reikalingas vienintelis parametras URL (kaip tekstas), kelias iki XML dokumentą, kurį norimą įkelti. Sekantis pavyzdys parodo kaip XML objektas įkelia dokumentą pavadintą duomenys.xml iš to paties katalogo, kuriame yra pats filmas:

```
var xml_obj:XML = new XML();
xml_obj.load("duomenys.xml");
```

Kelias iki XML dokumento gali būti pilnas URL. Pagal nutylėjimą ActionScript leidžia įkelti XML dokumentus tikrai iš to paties internetinio adreso, kuriame yra filmas. Taip yra padaryta dėl didesnio saugumo.

1.6.3. XML duomenų išsiuntimas

XML duomenų išsiuntimas yra ne mažiau svarbus negu duomenų paėmimas. Dažnai išsiuntimas ir įkėlimas yra naudojami kartu. Yra du būdai XML duomenų išsiuntimui. Metodas send() XML objekto duomenis konvertuoja į XML tekstą bei nusiunčia jį į nurodytu adresu (URL).

Metodas `sendAndLoad()` daro tą patį, ką ir `send()` metodas, bet jis dar laukia atsako iš serverio ir įkelia tuos duomenis atgal į Flash filmą.

Metodas `send()` reikalauja mažiausiai vieno parametro – teksto nurodančio URL, į kurį yra pageidaujama išsiųsti XML duomenis pavyzdžiui:

```
xmlDuomenys.send("failas.cgi");
```

Metodas `send()` taipogi leidžia nurodyti naršyklės langą, į kurį pageidaujama įkelti serverio atsaką. Pavyzdžiui, žemiau pateiktas pavyzdys nusiunčia XML duomenis į serverį ir tada atvaizduoja atsakymą naujame naršyklės lange:

```
xmlDuomenys.send("failas.cgi", "_blank");
```

Metodas `sendAndLoad()` reikalauja dviejų parametrų. Pirmas parametras yra URL į kurį pageidaujama nusiųsti duomenis. Antras parametras yra XML objektas, kuriame norima gauti atsakymą. Atsako XML objektas turi turėti nustatytą `onLoad()` įvykių apdorojimo metodą:

```
xmlDuomenys.sendAndLoad("failas.cgi", xmlLoader);
```

1.6.4. LoadVars klasė

Klasė `LoadVars` suteikia galimybę išsiųsti ir priimti kintamuosius į Flash programas. Valdymo sąsaja duomenų siuntimui ir priėmimui `LoadVars` klasėje yra labai panaši į valdymo sąsają išsiunčiant ir priimant duomenis su XML klase. Skirtumas tarp `LoadVars` ir XML objektų yra tas, kad XML objektas yra skirtas XML duomenims, o `LoadVars` klasė dirba su kintamasis-reikšmė duomenimis.

Norint pradėti naudotis `LoadVars` klase būtina sukurti `LoadVars` objektą. Sukūrimo būdas yra paprastas, reikia iškviešti konstruktoriaus funkciją. Funkcija nereikalauja jokių papildomų parametrų:

```
var lvObj:LoadVars = new LoadVars();
```

1.6.5. LoadVars duomenų įkėlimas

Duomenų įkėlimas pasinaudojant `LoadVars` objektais, veikimas labai panašus į duomenų įkėlimą naudojantis XML objektais. Pirmame žingsnyje reikia susikurti objektą. Vėliau reikia iškviešti `load()` metodą iš objekto, kaip parametą reikia nurodyti URL adresą, iš kur duomenis reikia pasiimti. Kaip ir su XML objektais įkėlimo procesas yra asinchroninis, taigi galima nurodyti `onLoad()` metodą, kuris iškviečiamas, kada duomenys yra įkelti (arba nepavyko įkelti).

Perduodami duomenys LoadVars objekte turi būti URLEncoded, kintamasis-reikšmė porų formatu. Kitais žodžiais kiekvienas kintamasis ir reikšmė turi būti atskirti lygybės simboliu (=), kiekviena kintamojo-reikšmės pora turi būti atskirta simboliu (&) ir visi nestandartiniai simboliai turi būti užkoduoti. Pavyzdys, tarkime, kad norime įkelti į Flash filmuką tekstą „vienas du trys“ ir „keturi penki“. Duomenys turėtų būti pateikti tokiu formatu:

```
pirmi_trys=vienas%20du%20trys&kiti=keturi%20penki
```

Duomenys gali būti patalpinti tekstiniame faile arba gali būti sugeneruoti serverio pusės skripto, bet abiem atvejais duomenys turi būti pateikiami tuo pačiu formatu.

1.6.6. LoadVars duomenų siuntimas

Kaip ir su XML objektais duomenys gali būti išsiųsti naudojantis LoadVars objektu su send() arba sendAndLoad() metodu. Kiekvienas metodas veikia išsiųsdamas objekte nustatytus kintamuosius į nurodytą URL. Abiem metodais kintamieji yra siunčiami POST metodu, jei nėra nurodytas kitas.

Metodas send() reikalauja tikrai vieno parametro: URL į kurį kintamasis turėtų būti nusiųstas. Pavyzdys:

```
var lvObj:LoadVars = new LoadVars();  
lvObj.kintA = 5;  
lvObj.kintB = "tekstas";  
lvObj.kintC = "tekstas 2";  
lvObj.send("http://www.adresas.lt/failas.php");
```

Metodas sendAndLoad() veikia labai panašiai kaip ir XML objekto sendAndLoad metodas. Jis reikalauja dviejų parametrų: URL į kurį yra siunčiami kintamieji ir LoadVars objekto, kuriame siuntimo rezultatai bus pateikti. Su send() metodu

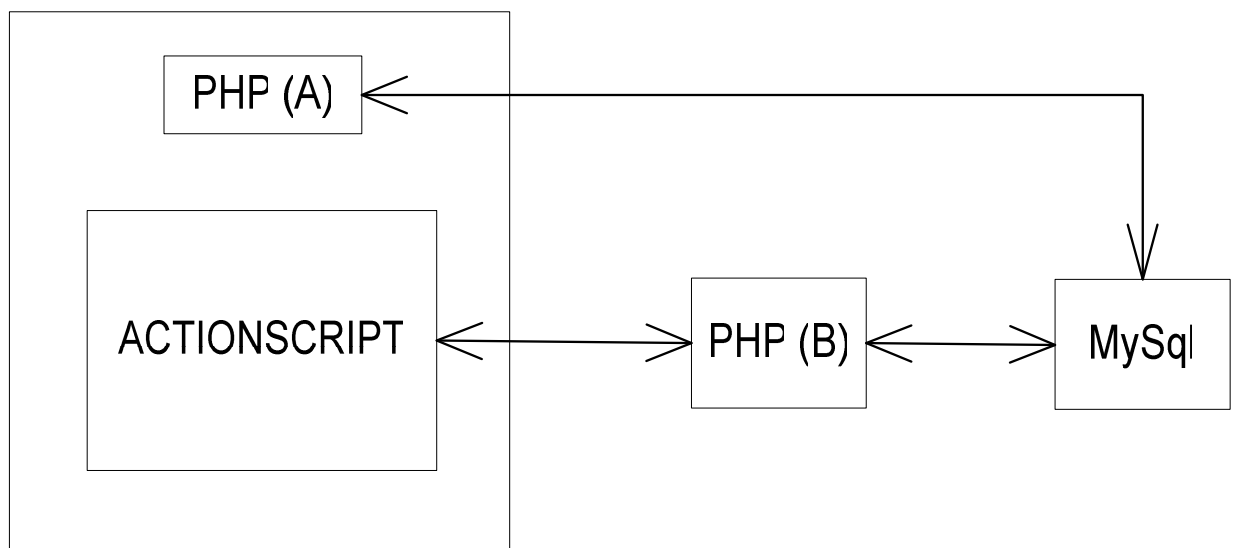
1.7. Derinių atvaizdavimas

Duomenų saugojimo procedūra yra daug paprastesnė už duomenų paėmimo procedūrą. Duomenų saugojimo procedūroje (žemiau pateiktas paveikslas) Dalyvauja objektai: ActionScript, PHP (B) ir MySQL.

Norint atlikti šią procedūrą reikia: dviejų PHP failų (A ir B), ActionScript objekto ir MySQL duomenų bazės. Norint duomenis paimti iš duomenų bazės susiduriama su keliomis

problemomis. Puslapis (PHP A), kuriame yra ActionScript objektas tarpusavyje neturi jokio ryšio. Jeigu būtų galimybė sudaryti kokį nors ryšį, būtų galima perduoti atitinkamus parametrus, su kuriais ActionScript objektas kreiptųsi į PHP B failą taip palengvinant duomenų paėmimą.

Kita problema yra ta, kad ActionScript objektas visada kreipiasi į vieną ir tą patį PHP B failą ir neišeina iš šio objekto perduoti GET ar POST metodu reikalingų kintamųjų. Jeigu duomenų bazėje būtų tikrai vienas derinys, tada viskas būtų labai paprasta, PHP B failas tokiu atveju galėtų būti paprasčiausias XHTML failas, bet derinių yra daug ir kiekvieną kartą reikia rodyti skirtingus derinius.



1.7.1. paveikslas. Derinio atvaizdavimo schema

Norint išspręsti šią duomenų perdavimo problemą duomenų bazėje reikėjo sukurti papildomą atributą, kuris būtų rodomo derinio identifikacinis numeris. Šis papildomas atributas nėra bendras visiems vartotojams, kiekvienas vartotojas turi jam skirtą atributą.

Taigi vartotojas pasirinkęs derinį, kuri nori pamatyti patenka į PHP A puslapį, kuris į duomenų bazę įrašo derinio identifikacinį numerį, kurį jisai nori pamatyti. Vėliau tame pačiame A puslapyje ActionScript objektas kreipiasi į PHP B failą su einamąja prisijungimo sesija. PHP B faile iš sesijos kintamųjų gavus reikalingą informaciją PHP B failas duoda užklausą atitinkamo vartotojo, kad gautų derinio numerį, kurio duomenis reikės perduoti ActionScript objektui. Sužinojus derinio numerį bei surinkus visa informaciją apie derinyje naudojamus elementus PHP B

failas sugeneruoja duomenų masyvą, kurį pasiima ActionScript objektas bei sėkmingai atvaizduoja derinį.

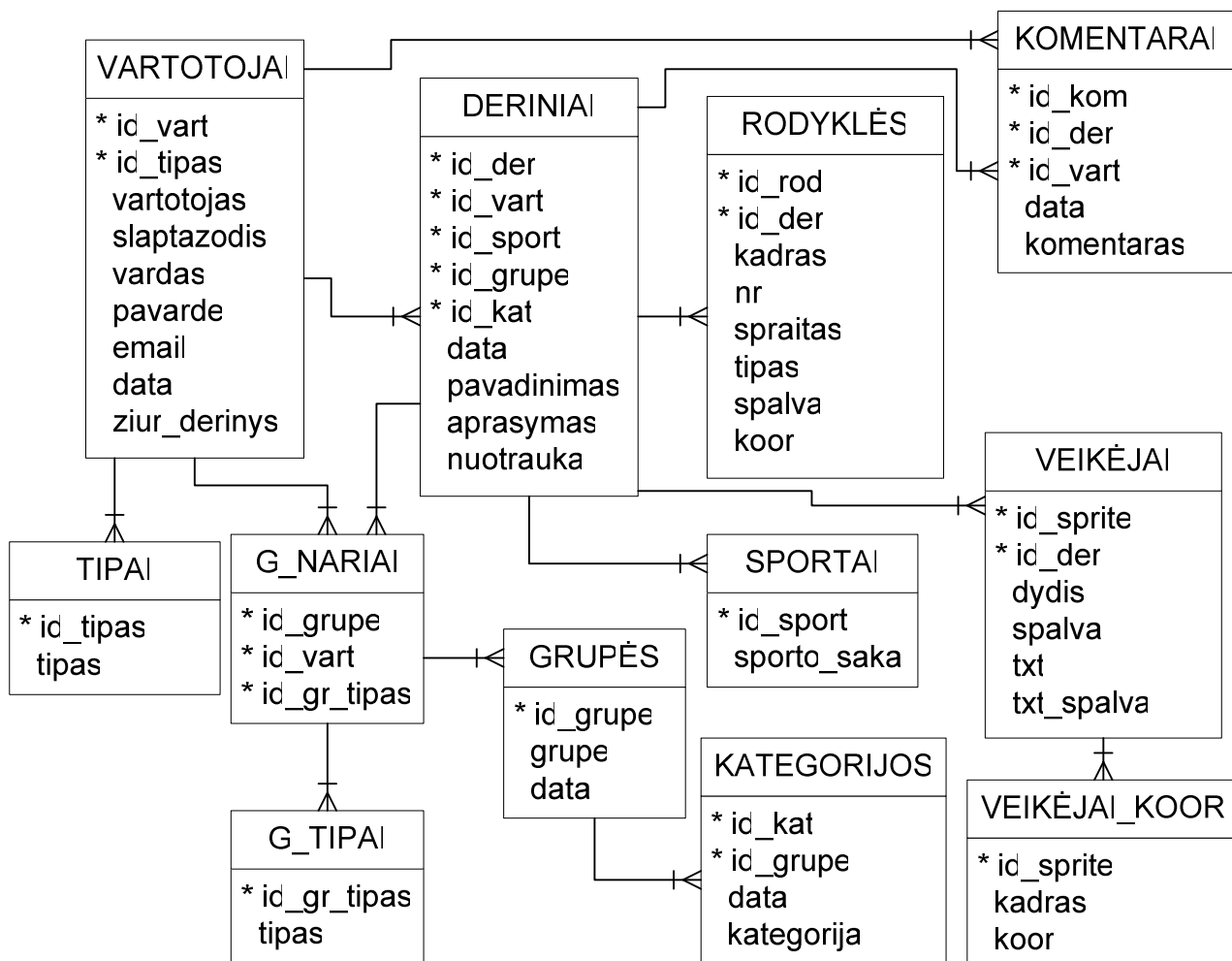
1.8. Apibendrinimas

- XML yra kalba su vartotojo sukurtais elementais, kurie suteikia kontekstą duomenims. Naudojant XML galima saugoti ir siųsti duomenis formatu, kuris nepriklauso nuo platformos, tuo pat metu išlaikant kintamuosius taip pat neprarandant jų prasmės universalioje suprantamoje struktūroje.
- XML savo struktūra yra panašus į HTML, bet daug griežtesnės taisyklės, kurių pagalba kalba tampa struktūrizuota. XML, kuris išlaiko visas taisykles, yra žinomas kaip taisyklingai suformuotas XML.
- Naudojant XML objektą, ActionScript pagalba galima sukurti išsiųsti priimti ir apdoroti XML dokumentus.
- LoadVars ir XML objektai duomenis išsiunčia ir priima tais pačiais būdais. Skirtumas tarp LoadVars ir XML objektų yra duomenų formatas. XML objektai skirti dirbti XML duomenimis, o LoadVars su duomenimis kintamasis-reikšmė duomenų poromis.
- Informacinėje sistemoje renkantis tarp GET ir POST metodų buvo pasirinktas POST metodas, nes šio metodo pagalba galima perduoti didesnę kiekį duomenų. Taipogi šis metodas yra saugesnis negu GET metodas, POST metodu yra paprasčiau perduoti lietuviškus rašmenis.
- Informacinėje sistemoje buvo pasirinkta naudoti LoadVars klasę nes naudojant savo sudarytą struktūrą galima sumažinti perduodamų duomenų kiekį lyginant su XML klase.
- Nėra galimybės iš ActionScript objekto duomenis tiesiogiai perduoti ar pasiimti iš MySQL duomenų bazės. Norint atlikti šią procedūrą reikalingas papildomas tarpininkas.
- Duomenų paėjimas iš MySQL duomenų bazės į ActionScript objektą yra panašus į AJAX veikimą.

2. PROJEKVINĖ DALIS

Informacinėje sistemoje realizuotoje duomenų bazėje nėra atsikartojančių laukų (trečia normalinė forma). Visi ryšiai tarp lentelių yra 1 su daug. Visos lentelės turi raktus.

Pavyzdžiui, sukurto derinio informacija su kuria yra atkuriami ir animuojami deriniai yra saugoma šiose lentelėse: deriniai, rodyklės, veikėjai, *veikėjai_koor*. Visos šios keturios lentelės turi bendrą identifikatorių *id_der*, kuris kiekvienam deriniui yra unikalus ir nesikartoja.

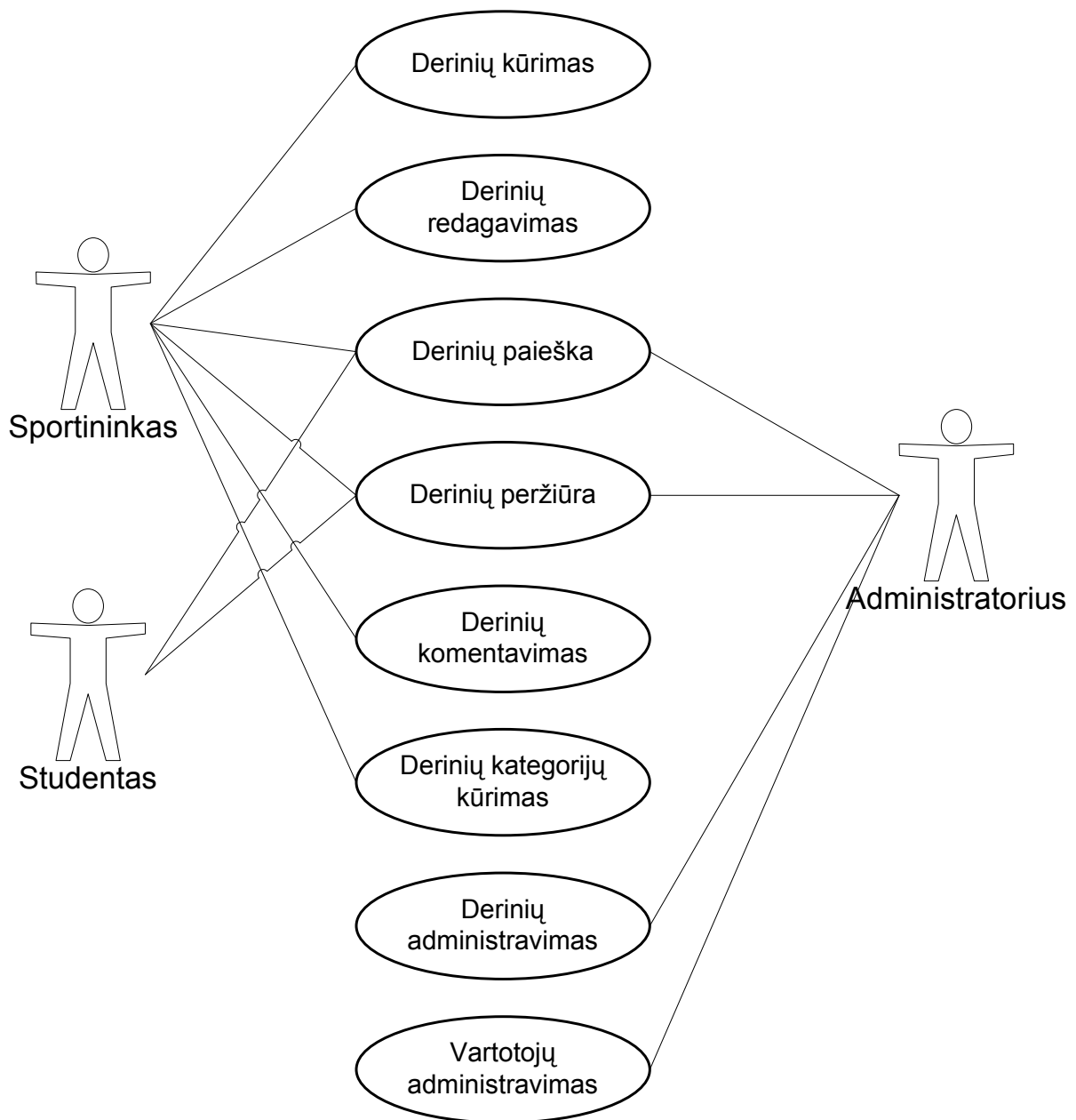


2.1. paveikslas. Duomenų bazės schema

2.1. lentelė. DB Esybių paaiškinimai

Esybė (lentelė)	Aprašymas
VARTOTOJAI	Informacija apie vartotojus.
TIPAI	Galimi IS vartotojų tipai.
G_NARIAI	Vartotojo narystės grupėse bei rolės grupėse kurioms priklauso vartotojas.
G_TIPAI	Galimi informacinės sistemos grupių vartotojų tipai.
DERINIAI	Derinių lentelė.
GRUPĖS	Vartotojų sukurtų grupių sąrašai.
RODYKLĖS	Deriniuose naudojamų rodyklių duomenų lentelė.
SPORTAI	Sporto šakos, kurių derinius galima kurti informacinės sistemos pagalba.
KATEGORIJS	Kategorijos, kurių pagalba vartotojai gali suskirstyti savo sukurtus derinius į savo sugalvotas kategorijas.
KOMENTARAI	Derinių komentarų lentelė.
VEIKĖJAI	Deriniuose esančių veikėjų lentelė.
VEIKĖJAI_KOOR	Deriniuose esančių veikėjų koordinačių lentelė.

2.1. Panaudojimo atvejai



2.1.1. paveikslas. Informacinės sistemos panaudojimo atvejai

2.2. Panaudojimo atvejų sąrašas

a) Derinių kūrimas

Tikslas: Suteikti galimybę sistemos vartotojams kurti komandinių sporto šakų (krepšinio, rankinio, futbolo, ledo ritulio) sportinius derinius.

Aktoriai: sportininkai.

Prieš-sąlygos: Būtina būti užsiregistravusiam sistemoje.

Sužadavimo sąlyga: Derinio kūrimo elementų komponavimas.

Po-sąlyga: Sukurtas derinys patalpinamas duomenų bazėje, vėliau jį gali peržiūrėti kiti grupei priklausantys nariai.

b) Derinių redagavimas

Tikslas: Suteikti galimybę sistemos vartotojams redaguoti jų pačių sukurtus derinius.

Aktoriai: sportininkai.

Prieš-sąlygos: Būtina būti užsiregistravusiam sistemoje, būtina turėti bent vieną savo sukurtą ir sistemoje esantį derinį.

Sužadavimo sąlyga: Derinio kūrimo elementų komponavimas.

Po-sąlyga: Modifikuotas derinys patalpinamas duomenų bazėje, vėliau jį gali peržiūrėti kiti grupei priklausantys nariai.

c) Derinių paieška

Tikslas: Suteikti galimybę sistemos lankytojams greitai ir patogiai surasti ieškomus derinius pagal įvairius paieškos kriterijus.

Aktoriai: sportininkai, studentai, administratoriai.

Prieš-sąlygos: Būtina būti užsiregistravusiam sistemoje, sistemos duomenų bazėje turi būti bent vienas derinys.

Sužadavimo sąlyga: Vartotojas turi nustatyti pageidaujamus paieškos kriterijus, kurių pagalba bus surasti visi vartotojui prieinami deriniai.

Po-sąlyga: Matomas pagal įvestus kriterijus rastų derinių sąrašas, kuriuos vartotojas gali peržiūrėti.

d) Derinių peržiūra

Tikslas: Suteikti galimybę sistemos vartotojams pamatyti savo, kitų sukurtus derinius.

Aktoriai: sportininkai, studentai, administratoriai.

Prieš-sąlygos: Būtina būti užsiregistravusiam sistemoje, sistemos duomenų bazėje turi būti bent vienas derinys.

Sužadavimo sąlyga: Derinio paleidimo mygtukas.

Po-sąlyga: Galima derinį pradėti žiūrėti iš naujo, skaityti po deriniu esančius komentarus, palikti savo komentarą apie konkretų derinį.

e) Derinių komentavimas

Tikslas: Suteikti galimybę sistemos vartotojams komentuoti (diskutuoti) pamatytus savo, kitų sukurtus derinius.

Aktoriai: sportininkai, studentai.

Prieš-sąlygos: Būtina būti užsiregistravusiam sistemoje, sistemos duomenų bazėje turi būti bent vienas derinys.

Sužadinimo sąlyga: Komentaro(ų) rašymas.

Po-sąlyga: Parašytas komentaras yra matomas visiems sistemos lankytojams, arba tik tai konkrečios grupės vartotojams, kuriai priklauso vartotojas.

e) Derinių kategorijų kūrimas

Tikslas: Suteikti galimybę sistemos vartotojams savo sukurtus derinius patalpinti savo sugalvotose kategorijose.

Aktoriai: sportininkai.

Prieš-sąlygos: Būtina būti užsiregistravusiam sistemoje, turi būti pradėtas derinio kūrimas.

Sužadinimo sąlyga: Prisijungus prie sistemos sukurti, kokį nors derinį, vėliau apipavidalinant derinį galima sukurti naują kategoriją arba pasirinkti esamą.

Po-sąlyga: Sukuriama nauja kategorija, kurioje jau būna vienas derinys ir kurioje galima patalpinti kitus derinius.

a) Derinių administravimas

Tikslas: Spręsti visas kylančias iškilusias problemas su deriniais.

Aktoriai: administratoriai.

Prieš-sąlygos: Turėti teisę prieiti prie MySQL duomenų bazės bei turėti galimybę atlikti pakeitimus.

Sužadinimo sąlyga: Problemų identifikavimas.

Po-sąlyga: Išsprendžiamos iškilusios problemos.

a) Vartotojų administravimas

Tikslas: Sistemoje esančių vartotojų priežiūra.

Aktoriai: administratoriai.

Prieš-sąlygos: Turėti teisę prieiti prie MySQL duomenų bazės bei turėti galimybę atlikti pakeitimus.

Sužadavimo sąlyga: Problemų identifikavimas.

Po-sąlyga: Išsprendžiamos iškilusios problemos.

2.3. Aktorių funkcijos

2.3.1. Sportininkas

Šis aktorius turi galimybę, kurti naujus žaidimų derinius. Kuriant leidžiama pasirinkti pageidaujamą sporto šaką: futbolas, krepšinis, rankinis, ledo ritulys ir t.t. Kuriant derinius galima pasirinkti įvairius derinių atvaizdavimo elementus (linijos, trumpi komentarai, sričių pažymėjimas, žaidėjų spalvų pasirinkimas). Galima pasirinkti vienos ir kitos komandos žaidėjų spalvas, žaidėjams galima suteikti numerius, išskirti pagrindinius žaidėjus, braižyti įvairias pagalbines linijas, įvairiose aikštelės vietose sukurti komentarus. Sukurtus derinius galima, bet kuriuo momentu redaguoti, redaguoti bus galima, kai sistema bus papildyta, atnaujinta: naujais, patobulintais derinių atvaizdavimo elementais. Derinių kūrimo ir redagavimo funkcijos bus suteikiamos tik *sportininkams*.

Sukurtus derinius bus galima, bet kuriuo laiko momentu peržiūrėti. Peržiūrėti bus galima tik dviejų tipų derinius: komandos deriniai ir laisvai publikuojami deriniai. Derinius peržiūrėti bus galima tiksliai atlikus derinių paieškos procedūrą. Peržiūrint derinius bus galima juos komentuoti. Taip pasakyti savo nuomonę apie konkretų derinį. Peržiūrint derinius pasirinktu laiko momentu ir sustabdžius derinį bus galima suformuoti užfiksuotą vaizdą, bei jį atsispausdinti arba išsaugoti konkretaus komentaro nuotrauką.

2.3.2. Studentas

Šiuo aktoriumi galės tapti, bet koks sporto entuziastas nemokamai. Šios kategorijos vartotojai galės atlikti tą patį, ką gali atlikti *sportininkas* tik negalės kurti ir redaguoti derinių. *Studentas* galės peržiūrėti tiksliai vienos srities derinius: laisvai prieinamus, tokius derinius gali padaryti *sportininkas*. Tarkim konkreti komanda turi derinį, kuris yra paprastas ir jai ne itin svarbus, todėl gali jį pažymėti kaip laisvai prieinamą. Nuo tokio momento šis derinys bus prieinamas visiems *studentams* ir *sportininkams*.

Šio tipo vartotojams nebus suteikta teisių kurti derinių todėl, kad taip bus siekiama išlaikyti aukštą derinių kokybę. Bus siekiama apsaugoti nuo menkos kokybės įvairių *studentų* derinių.

2.3.3. Administratorius

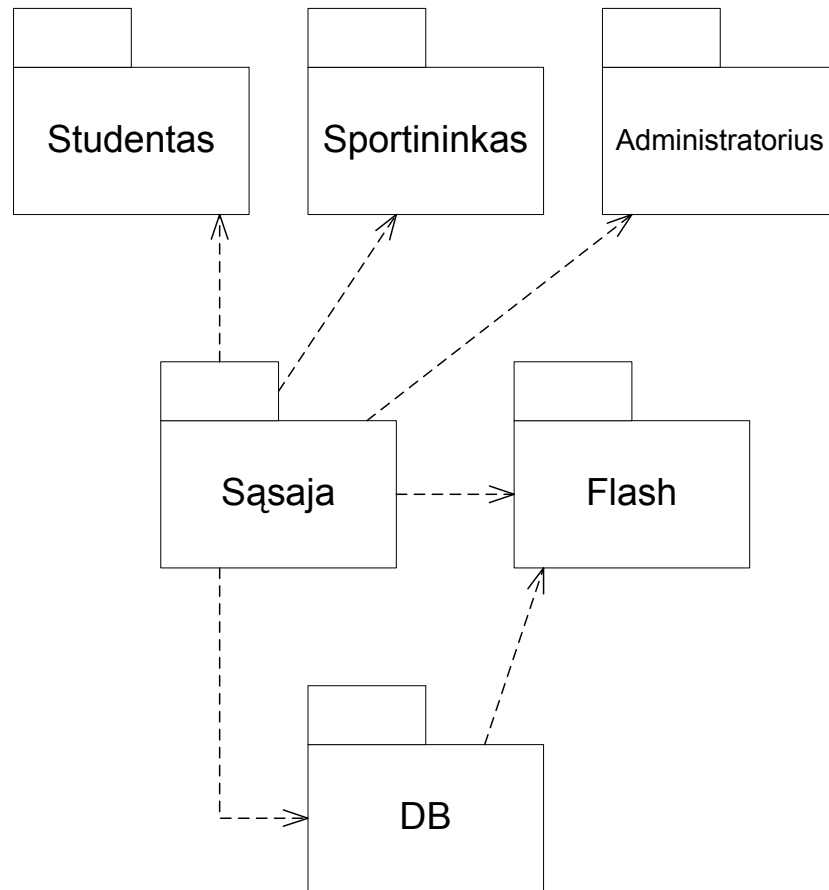
Šio aktorius pagrindinė funkcija bus esamų derinių bei jų komentarų priežiūra. Jeigu sukurtuose deriniuose bus naudojami kaip vaizdinės priemonės įvairūs žodžiai tai tie žodžiai negalės būti neetiški, įžeidžiantys kitus asmenis ir t.t. Pačiuose deriniuose žodinių elementų labai daug nebūna, todėl pagrindinis aktorius dėmesys bus skiriamas derinių komentarams.

Jeigu konkretus asmuo įžeidžia kitą asmenį, naudoja neetiškus žodžius, ar kitaip prasižengia, gauna nuo šio aktorius įspėjimą. Už antrą įspėjimą yra blokuojamas priėjimas prie sistemos. Už antrą įspėjimą bus blokuojamas priėjimas ne tik *studentams*, bet ir *sportininkams*. Tam, kad *sportininkas* galėtų atgauti priėjimą prie sistemos turės sumokėti piniginę baudą. *Studentams* priėjimas prie sistemos bus panaikintas visiems laikams.

Aktorius taipogi lauks įvairiausių nusiskundimų, pageidavimų, pasiūlymų iš *sportininkų, studentų*.

2.4. Loginis vaizdas

Sistemos suskaidymas į paketus aukščiausiam lygmenyje:



2.4.1. paveikslas. Sistemos išskaidymas į paketus

Suskaitytoje sistemoje gaunasi trys informacinės sistemos vartotojai, bei trys sistemos komponentai. Priėjimas prie duomenų bazės duomenų ir Flash (ActionScript) objekto vyksta su sąsajos pagalba.

Norint, kad sistema sėkmingai funkcionuotų turi būti bent vienas sistemos vartotojas (studentas, sportininkas, administratorius) ir būtinai sąsajos (PHP), Flash (ActionScript) ir duomenų bazės (MySQL) komponentai.

2.5. Funkciniai reikalavimai

- **Vartotojų registravimas.** Informacinė sistema turi turėti galimybę užregistruoti naujus sistemos vartotojus, bei padėti vartotojams, kurie dėl vienokių ar kitokių priežasčių negali prisijungti prie sistemos.
- **Derinių kūrimas.** Galimybė kurti krepšinio, futbolo, rankinio, ledo ritulio sporto šakų derinius animuotus derinius. Galimybė kurti pageidaujamo dydžio, spalvų, aktorius. Numeruojant aktorius naudoti ne tik skaičius, bet ir kitus simbolius, lietuviškus rašmenis. Deriniuose galima įterpti pageidaujamas rodykles. Rodykles rinktis iš kelių tipų, rodykles sudarant iš pasirinkto kiekio atkarpų, kurių jungiamuosius tašus būtų galima nesunkiai valdyti. Rodyklės gali prasidėti vieno tipo (pavyzdžiui, tiesi linija), o baigtis (pavyzdžiui, brūkšninė linija). Derinių kūrimas turi būti lengvai suprantamas, neperkrautas.
- **Derinių redagavimas.** Vartotojai sukūrę savo derinius turi turėti galimybę redaguoti jau sukurtus ir duomenų bazėje patalpintus derinius.
- **Grupių sukūrimas.** Prisiregistravę vartotojai gali sukurti naują grupę arba pateikti prašymą kitų grupių administratoriams dėl priėmimo į egzistuojančias grupes. Vartotojas sukūręs naują grupę tampa jos administratoriumi.
- **Galimybė priklausyti kelioms grupėms.** Realizuota duomenų bazės architektūra turi būti pritaikyta tokiems atvejams, kada vartotojas gali priklausyti kelioms grupėms. Skirtingose grupėse vartotojo teisės gali skirtis.
- **Derinių paieška.** Atlikti pageidaujamų derinių paiešką pagal užsiduotus paieškos nustatymus. Paiešką turi būti galimybė atlikti pasirinktame laiko intervale. Jeigu vartotojas priklauso kelioms grupėms, pasirinkti kurios nors vienos grupės derinių paiešką ar ieškoti derinių visose grupėse, kurioms priklauso vartotojas.
- **Derinių peržiūra.** Peržiūrimas derinio duomenys ActionScript objektui turi būti perduodami kaip įmanoma greičiau, taip greitai gaunant galimybę žiūrėti derinį. Derinį galima stebėti po vieną kadrą arba visą iš karto. Galima rinktis pageidaujamą derinio žiūrėjimo greitį, atvaizduoti rodykles, kur sekančiame ėjime bus visi aktoriai arba ankstesniame ėjime buvo visi aktoriai.
- **Derinių šalinimas.** Derinius šalinti gali tik grupių administratoriai arba derinį sukūręs asmenys. Šalinant derinį turi būti pašalinta visa su deriniu susijusi informacija: aktoriai, aktorių koordinatės, rodyklės.
- **Derinių komentavimas.** Patekus į derinio peržiūros puslapį turi būti galimybė palikti įspūdžius, nuomonę apie peržiūrėtą derinį.

2.6. Nefunkciniai reikalavimai

2.6.1. Reikalavimai sistemos išvaizdai

- Derinio objektas turi tilpti 800x600 skiriamosios gebos lange.
- Sąsaja turi būti kaip įmanoma paprastesnė, neperkrauta, neatstumianti, patraukti.
- Apsauga nuo neteisingų duomenų įvedimo, bei nuo kitokių atsitiktinių, nepageidautinų vartotojo veiksmų.

2.6.2. Reikalavimai panaudojamumui

- Apsauga nuo neteisingų duomenų įvedimo, bei nuo kitokių atsitiktinių, nepageidautinų vartotojo veiksmų.
- Sistema turi suteikti galimybę naudotis net ir labai mažai patyrusiems vartotojams.

2.6.3. Reikalavimai vykdymo charakteristikoms

- Duomenų bazė turi būti sukurta taip, kad esant dideliame duomenų kiekiui būtų galima kaip įmanoma greičiau gauti ieškomų derinių sąrašą.

2.6.4. Reikalavimai veikimo sąlygoms

- Turi būti galimybė prieiti prie sistemos iš bet, kurios vietos, kur yra interneto prieiga.

2.6.5. Reikalavimai sistemos priežiūrai

- Sistema turi būti kaip įmanoma labiau ištobulinta, kad kuo daugiau darbų būtų galima atlikti prisijungus prie pačios sistemos (nelendant į išėities kodą).

2.6.6. Reikalavimai saugumui

- Vartotojo duomenys turi būti prieinami tik tai sutikus konkrečiam vartotojui.
- Jeigu yra sukurta grupė, tai jos derinių kiti vartotojai pamatyti negali.

2.6.7. Kultūriniai/politiniai reikalavimai

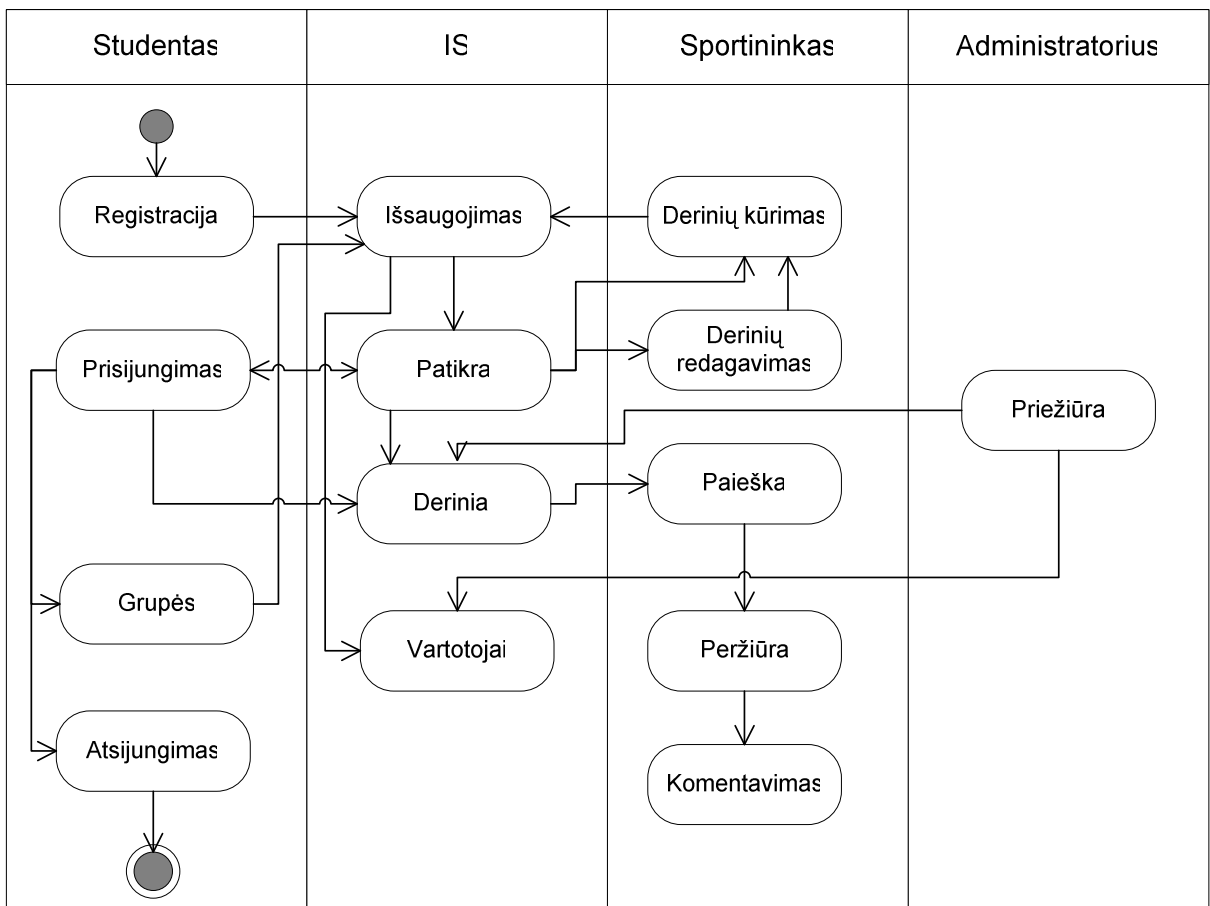
- Sistemoje naudojama lietuvių kalba, terminai turi būti taisyklingi, nenaudoti jokių svetimybės žodžių.

2.6.8. Teisiniai reikalavimai

- Sistema turi atitikti egzistuojančias, priimtas teises normas toje valstybėje, kurioje bus informacinės sistemos serveris.

2.7. Veiklos diagrama

Veiklos diagramoje galima pamatyti kaip žmogiškieji aktoriai bendrauja su IS:



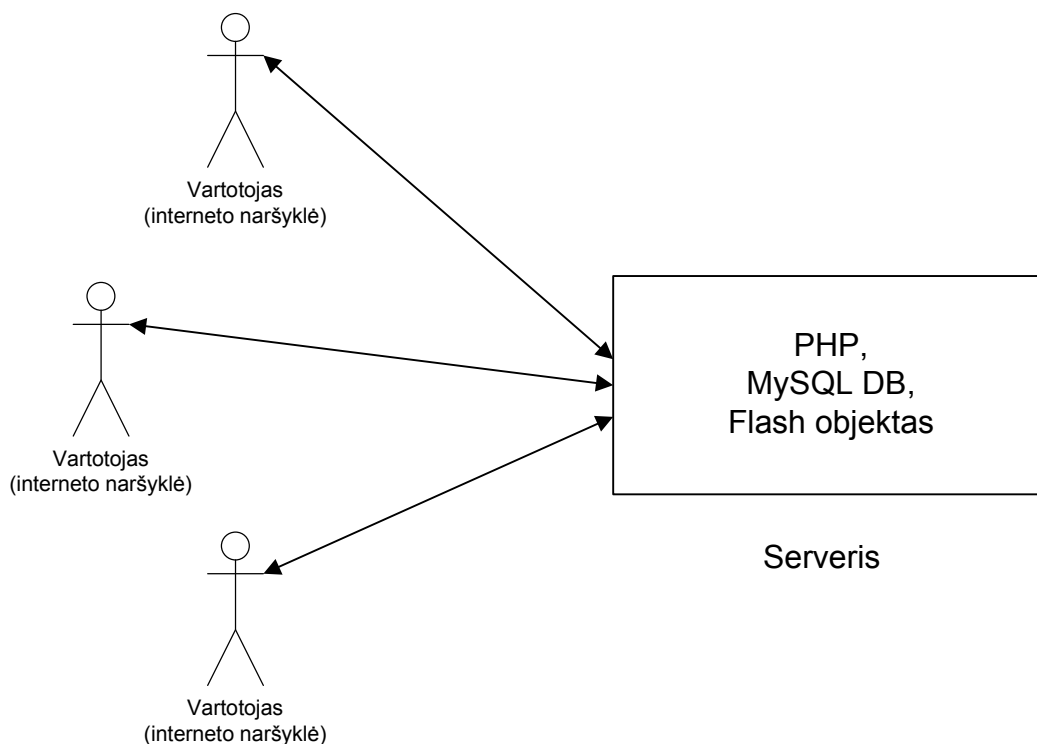
2.7.1. paveikslas. Veiklos diagrama

Informacinę sistemą sudaro šie aktoriai:

- Studentas.
- IS (informacinė sistema).

- Sportininkas.
- Administratorius.

2.8. Sistemos išdėstymo vaizdas



2.8.1. paveikslas. Išdėstymo diagrama

Sistemos išdėstymo diagramoje parodyta, kad sistemos vartotojai su sistema bendrauja interneto naršyklės pagalba. Informacinė sistema sudaro vienas serveris, kuriame turi būti šie komponentai:

- Apache (HTTP servisas)
- MySQL
- PHP

Šie trys komponentai yra būtina. Jei neveiktų bent vienas iš šių komponentų sistema negali funkcionuoti.

Jeigu ateityje labai padidėtų sistemos vartotojų skaičius ir sulėtėtų sistemos darbas būtų galima apsvarstyti MySQL duomenų bazės perkėlimą į atskirą serverį.

2.9. Serverio reikalavimai

Rekomenduojami reikalavimai serveriui, į kurį būtų galima įdiegti informacinę sistemą:

2.9.1. lentelė. Reikalavimai serveriui

Techninė įranga	
Procesorius	>= 1 Ghz
Atmintis	>= 1 Gb
Kietasis diskas	>= 30 Gb
Programinė įranga	
Operacinė sistema	Linux, BSD, Windows
Tinklo serveris	Apache
Programavimo kalba	PHP
Duomenų bazė	MySQL

2.10. Vartotojo kompiuterio reikalavimai

Rekomenduojami reikalavimai vartotojo kompiuteriui:

2.10.1. lentelė. Reikalavimai klientui

Techninė įranga	
Procesorius	>= 1.5 Ghz
Atmintis	>= 256 Mb
Kietasis diskas	>= 20 Gb
Vaizdo plokštė	>= 16 Mb
Monitoriaus skiriamoji geba	>= 1024x768
Programinė įranga	
Operacinė sistema	Linux, BSD, Windows šeimos OS
Interneto naršyklė	Internet Explorer, Opera, Firefox
Kitkas	Flash filmų leistuvai

3. EKSPERIMENTINĖ DALIS

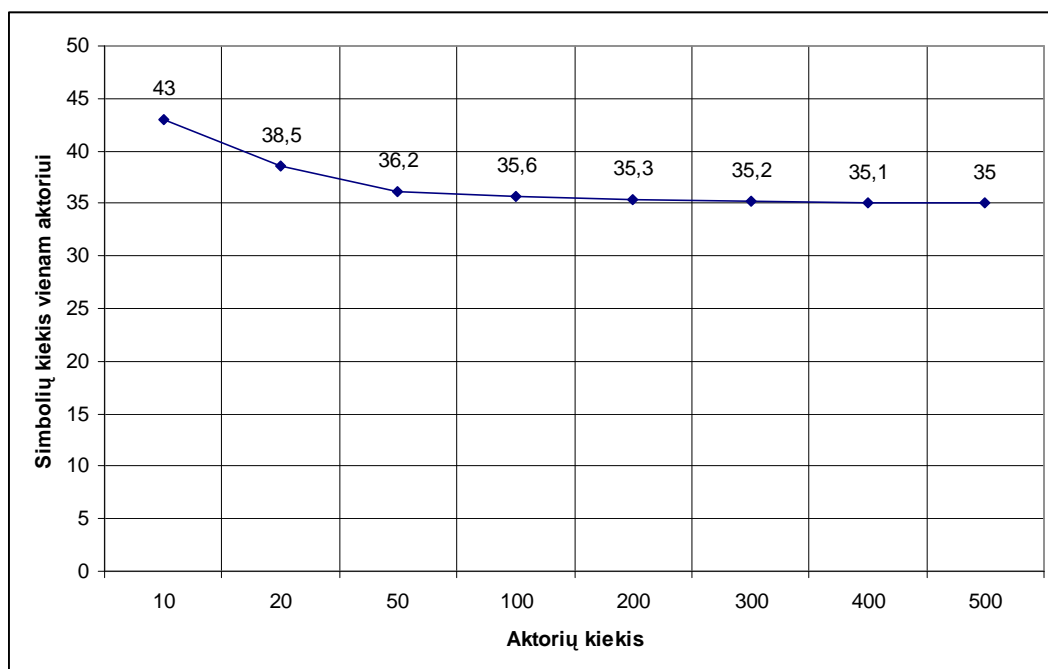
3.1. Eksperimentų tikslas

- Paanalizuoti duomenų perdavimo kiekius.
- Išbandyti kokį aktorių skaičių ActionScript objektas gali sėkmingai animuoti.

3.2. Duomenų perdavimo eksperimentas su nejudančiais aktoriais

Šiame eksperimente buvo sugeneruoti testai, kurių pagalba buvo išmatuotas vidutinis simbolių kiekis vienam aktoriui, kurį ActionScript objektas turi pasiimti iš (perduoti) PHP sugeneruoto failo. Šio eksperimento atveju nejuda nei vienas aktorius (nėra nei vieno kadro, derinys be animacijos).

Eksperimentuojama buvo pradėdant 10 aktorių ir baigiant 500 aktorių. Visi aktoriai eksperimento metu turėjo vienodą spalvą, dydį.

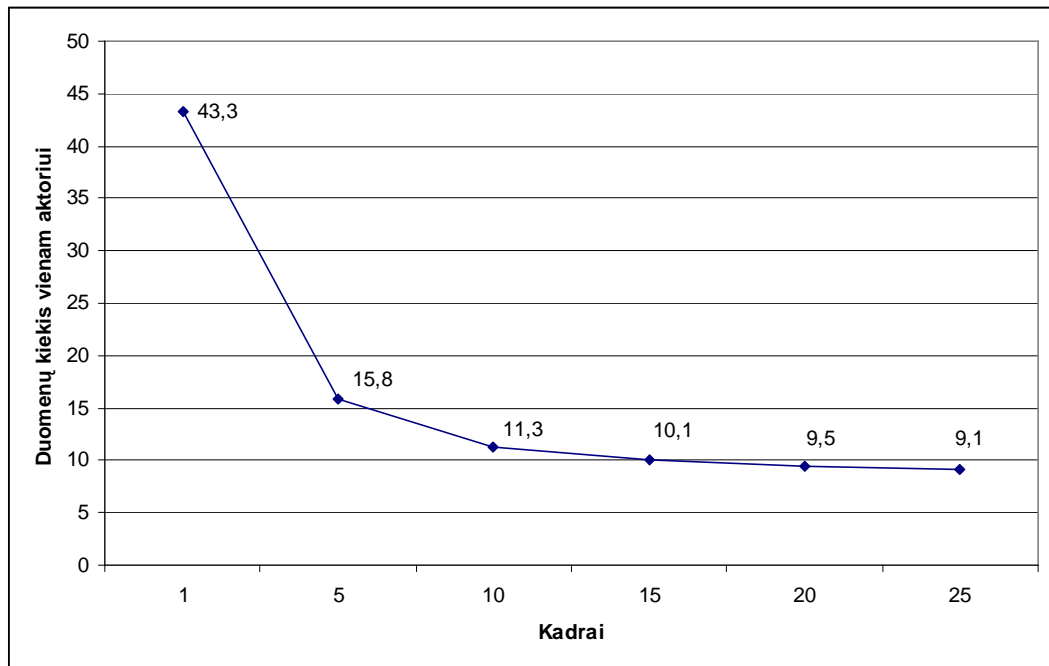


3.2.1. paveikslas. Duomenų kiekio nuo aktorių skaičiaus priklausomybė

Rezultatai parodė, kad vidutinis simbolių kiekis tenkantis vienam aktoriui mažėja didėjant aktorių kiekiui.

3.3. Duomenų perdavimo eksperimentas su judančiais aktoriais

Šio eksperimento atveju buvo apskaičiuojamas perduodamų duomenų kiekis vienam judančiam aktoriui. Buvo eksperimentuojama pradėdant 1 kadru ir baigiant 25 kadrais. Visuose (šešiuose) eksperimentuose buvo perduodamos 100 aktorių judėjimo koordinatės:



3.3.1. paveikslas. Duomenų kiekio nuo kadro skaičiaus priklausomybė

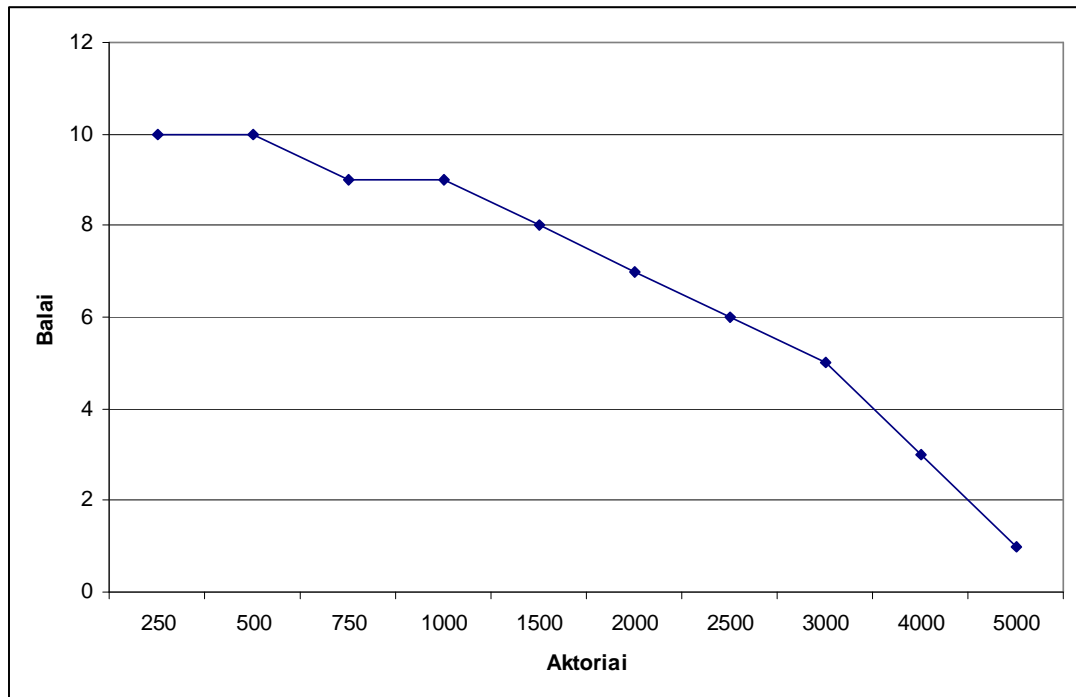
Iš aukščiau pateikto grafiko galima pastebėti, kad kuo animacijoje yra daugiau duomenų tuo mažiau duomenų tenka perduoti vienam aktoriui. Visais eksperimento atvejais ActionScript objekto greیتaveika buvo labai gera, animuojami objektai judėjo tolygiai.

Perduodamos koordinatės yra sveikieji skaičiai dviženkliai arba triženkliai. Eksperimento metu visi aktoriai buvo vienodo dydžio ir spalvos.

Jeigu toks duomenų perdavimas būtų realizuotas naudojant XML struktūrą, perduodamas duomenų kiekis būtų daugiau nei dvigubai didesnis. Taigi realizuota duomenų perdavimo struktūra leido minimizuoti perduodamų duomenų kiekį, taip pasiekiant didesnę informacinės sistemos greitį, taipogi taupomi interneto resursai.

3.4. Animuojamų objektų greitaveikos eksperimentas

Šiuo eksperimentu buvo bandoma įvertinti su koki kieki aktorių vienu metu gali animuoti ActionScript objektas. Testo metu buvo pasirinkta 10 balų greitaveikos vertinimo skalė. 10 balų reiškia, kad visi animuojami objektai juda sklandžiai be jokių nesklandumų, mažesnių balų atveju juntamas judamų objektų sutrikimas.



3.4.1. paveikslas. Aktorių kiekio animacijos vertinimas

Eksperimento rezultatai parodė, kad animuojant iki 500 aktorių nejuntami jokie ActionScript objekte esančių aktorių judesių netikslumai. Aktorių kiekiui padidėjus virš 500 juntami aktorių judesių netikslumai. Judesiai tampa netolygūs. Atlikus eksperimentą su 5000 tūkstančiai judančių aktorių, jų judėjimo nebuvo galima pavadinti judėjimu, judesių kokybė buvo labai bloga.

3.5. Eksperimentų apibendrinimas

- ActionScript objekto reakcijos atsako laikas yra geras, kai aktorių skaičius neviršija 1000 animuojamų aktorių skaičiaus.

- Panaudota duomenų perdavimo struktūra padeda sumažinti perduodamų duomenų kiekį lyginant su XML struktūra.

4. IŠVADOS

- Informacinėje sistemoje buvo sėkmingai realizuotas duomenų perdavimas iš ActionScript objekto į MySQL duomenų bazę bei iš duomenų bazės į ActionScript objektą.
- Informacinę sistemą pavyko sukurti universalią kelioms sporto šakoms, sistemą galima nesunkiai papildyti naujomis sporto šakomis.
- Sistemoje buvo atliktas svarbiausių metodų automatinis testavimas, kuris buvo atliekamas baltos ir juodos dėžės metodais.
- Informacinėje sistemoje buvo realizuotos visos pagrindinės užsakovo pradžiniuose reikalavimuose nurodytos sistemos funkcijos.
- Realizuota sistema galima naudotis, iš bet kurios pasaulio vietos, kurioje yra interneto prieiga, kas yra labai aktualu, kai sportininkams labai dažnai tenka keliauti po įvairias pasaulio šalis.
- Informacinės sistemos diegimas yra nesudėtingas, paprastas, reikalaujantis minimalių žmogaus pastangų.

5. LITERATŪROS SARAŠAS

- [1] XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition), <http://www.w3.org/TR/xhtml1/> [Žiūrėta 2009 05 15], August 2002.
- [2] **Samuel Hsieh**, Software engineering for Web application development, http://portal.acm.org/ft_gateway.cfm?id=948740&type=pdf&coll=Portal&dl=GUIDE&CFID=5942709&CFTOKEN=72405823, [Žiūrėta 2008 11 23], October 2003.
- [3] **Jason Gerner, Elizabeth Naramore, Morgan L. Owens, Matt Warden**, Linux, Apache, MySQL, and PHP5 Web Development, July 2006, ISBN-13: 978-0-7645-9723-7, p. 77-92.
- [4] **O'Reilly**, MySQL in a Nutshell, May 2005, ISBN: 0-596-00789-2, p. 30-33.
- [5] **Piero Fraternali, Paolo Paolini**, Model-driven development of Web applications: the AutoWeb system, http://portal.acm.org/ft_gateway.cfm?id=358110&type=pdf&coll=Portal&dl=GUIDE&CFID=5942709&CFTOKEN=72405823, [Žiūrėta 2009 04 02], October 2000.
- [6] **Alex Davies, Harrison Fisk**, MySQL® Clustering, March 2006, ISBN-10: 0-672-32855-0, chapter 3.
- [7] **Julie C. Meloni**, Sams Teach Yourself PHP, MySQL and Apache All in One, December 2003, ISBN: 0-672-32620-5, p. 356-362.
- [8] **Consortium for Computing Sciences in Colleges**, MySQL, PostgreSQL, and PHP: open source technologies for a database management course [Žiūrėta 2009 02 15], http://portal.acm.org/ft_gateway.cfm?id=775358&type=pdf&coll=Portal&dl=GUIDE&CFID=5922402&CFTOKEN=25191707
- [9] **Sams Publishing**, MySQL® Administrator's Guide, July 2004, ISBN: 0-672-32634-5, chapter 4.
- [10] **Andi Gutmans, Stig Sæther Bakken and Derick Rethans**, PHP 5 Power Programming, 2005, ISBN 0-131-47149-X, p. 412-415, p. 429-439.
- [11] **Vikram Vaswani**, How to Do Everything With PHP and MySQL, April 2005, ISBN: 0-07-225795-4, p. 150-172.
- [12] **Addison Wesley Professional**, PHP 5, April 2005, ISBN: 0-13-149862-2, p. 50-59.
- [13] **John Wiley & Sons**, Macromedia Flash 8 Bible, 2006, ISBN:0471746762, chapter 11.
- [14] **Keith Peters**, ActionScript 3.0 Animation, 2007, ISBN-13 (pbk): 978-1-59059-791-0,

chapter 7.

- [15] **Robert Hoekman, Rich Shupe**, Flash 8: Projects for Learning Animation and Interactivity, March 2006, ISBN-10: 0-596-10223-2, chapter 8.
- [16] **Sandro M. Corsaro**, Web production creativity: Flash finally hits the big screen,
http://portal.acm.org/ft_gateway.cfm?id=965360&type=pdf&coll=Portal&dl=GUIDE&CFID=5942709&CFTOKEN=72405823, [Žiūrėta 2009 01 25], July 2003.
- [17] **Joey Lott, Darron Schall, Keith Peters**, ActionScript 3.0 Cookbook, October 2006, ISBN: 0-596-52695-4, p. 48-70.
- [18] **Gyu Sang Choi, Jin-Ha Kim, Deniz Ersoz, Chita R. Das**, Architecture and implementation of Web sites: A multi-threaded PIPELINED Web server architecture for SMP/SoC machines,
http://portal.acm.org/ft_gateway.cfm?id=1060851&type=pdf&coll=Portal&dl=GUIDE&CFID=5942709&CFTOKEN=72405823, [Žiūrėta 2009 05 01], May 2005.
- [19] **Paolo Merialdo, Paolo Atzeni, Giansalvatore Mecca**, Design and development of data-intensive web sites: The Araneus approach,
http://portal.acm.org/ft_gateway.cfm?id=643480&type=pdf&coll=Portal&dl=GUIDE&CFID=5942709&CFTOKEN=72405823, [Žiūrėta 2009 05 10], February 2003.
- [20] **Frank S. Rietta**, Database systems II: Application layer intrusion detection for SQL injection,
http://portal.acm.org/ft_gateway.cfm?id=1185564&type=pdf&coll=Portal&dl=GUIDE&CFID=5942709&CFTOKEN=72405823, [Žiūrėta 2009 03 20], March 2006.
- [21] **Chris Snyder, Michael Southwell**, Pro PHP security, 2005, ISBN (pbk): 1-59059-508-4, p. 278-290.

6. TERMINŲ IR SANTRUMPŲ ŽODYNAS

UML	- Unifikuota modeliavimo kalba (Unified Modeling Language).
XML	- Išplečiama žymų kalba (Extensible markup language).
XHTML	- Išplečiama hiperteksto žymėjimo kalba (eXtensible HyperText Markup Language).
TCP/IP	- Tinklo protokolas, duomenų perdavimui.
HTTP	- Hiperteksto žymėjimo kalba.
DB	- Duomenų bazė.
PA	- Panaudojimo atvejis.
IS	- Informacinė sistema.
ActionScript	- Media programavimo kalba
SQL	- Struktūrizuota užklausų kalba.
MySQL	- Populiari atvirojo kodo reliacinių duomenų bazių valdymo sistema, dirbanti SQL kalbos pagrindu.
Apache	- Interneto (HTTP) servisas.
PHP	- Plačiai paplitusi dinaminė interpretuojama programavimo kalba, sukurta 1997 m. ir specialiai pritaikyta interneto svetainių kūrimui.

7. PRIEDAI

0/0

Aktoriaus parametrai:
 10 3

įkelti aktorių

Rodyklės:
R

Fono nuotrauka:
Krepšinis

Kurti kadra Trinti kadra

Ankstesn. ėjimo rodyklės:
AR

Sekančio ėjimo rodyklės:
SR

Greitis: 10

<< < Amin. > >> viso peržiūra Saugoti

7.1. paveikslas. Derinių kūrimo objekto pavyzdys