

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Nerijus Černiauskas

**Delninio kompiuterio programinės įrangos
projektavimo metodų tyrimas**

Magistro darbas

Darbo vadovas
doc. dr. E. Karčiauskas

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Nerijus Černiauskas

**Delninio kompiuterio programinės įrangos
projektavimo metodų tyrimas**

Magistro darbas

Kalbos konsultantė

Lietuvių k. katedros doc.

dr. J. Mikelionienė

2006 05

Vadovas

doc. dr. E. Karčiauskas

2006 05

Recenzentas

doc. dr. A. Riškus

2006 05

Atliko

IFM-0/2 gr. stud.

Nerijus Černiauskas

2006 05

Kaunas, 2006

TURINYS

1.	ĮVADAS	8
2.	ANALIZĖ	10
2.1	GALIMI SPRENDIMAI	10
2.1.1	<i>Ribotas užduočių atlikimo greitis</i>	10
2.1.2	<i>Vartotojo sąsajos problemos</i>	16
2.1.3	<i>Kintančios aplinkos problema</i>	22
2.2	TECHNOLOGIJŲ ANALIZĖ	23
2.2.1	<i>Microsoft .NET technologijos</i>	23
2.2.2	<i>Microsoft Windows 32-BIT API galimybės</i>	28
2.2.3	<i>Microsoft Windows 32-BIT API ir .NET Compact Framework palyginimas</i>	30
2.3	PANAŠIŲ SISTEMŲ APŽVALGA	31
2.4	PRIIMTO SPRENDIMO PAGRINDIMAS	35
3.	PROJEKTINĖ DALIS	36
3.1	BENDRA APŽVALGA	36
3.2	ARCHITEKTŪRINIS VAIZDAS	36
3.3	ARCHITEKTŪROS TIKSLAI IR APRIBOJIMAI	37
3.4	SISTEMOS STATINIS VAIZDAS	38
3.4.1	<i>Apžvalga</i>	38
3.4.2	<i>Paketų detalizavimas</i>	38
3.4.3	<i>Sistemos dinaminis vaizdas</i>	40
3.5	DUOMENŲ VAIZDAS	43
3.6	IŠDĖSTYMO VAIZDAS	44
4.	EKSPERIMENTINIS SISTEMOS TYRIMAS	45
4.1	EKSPERIMENTO TIKSLAS IR UŽDAVINIAI	45
4.2	EKSPERIMENTO OBJEKTAS	45
4.3	EKSPERIMENTINIO TYRIMO EIGA	45
4.4	KOKYBĖS KRITERIJŲ VERTINIMAS PAGAL ISO/IEC 9126 KOKYBĖS MODELĮ	49

4.4.1	<i>Funkcionalumas</i>	49
4.4.2	<i>Patikimumas</i>	49
4.4.3	<i>Panaudojamumas</i>	50
4.4.4	<i>Efektyvumas</i>	50
4.4.5	<i>Pernešamumas</i>	50
4.4.6	<i>Priderinamumas</i>	50
5.	IŠVADOS	51
6.	LITERATŪRA	52
7.	TERMINŲ IR SANTRUMPŲ ŽODYNAS	54
8.	PRIEDAI	55

SUMMARY

While there has been much successful work in developing rules to guide the design and implementation of interfaces for desktop machines and their applications, the design of mobile device interfaces is still relatively unexplored and unproven. This paper discusses the characteristics and limitations of current mobile device interfaces. Using existing interface guidelines as a starting point, a set of practical design guidelines for mobile device interfaces is proposed.

An experimental interface was created that ran on a Pocket PC 2002 mobile computer and used a simple events registration style interface to enter data of basketball's competition. The buttons of the registration system were changed in size between 12x12, 16x16 and 20x20 pixels and used a sound. Results showed that sounds significantly improved usability for both standard and small buttons.

Paveikslukų sąrašas

1 pav.	Kliento – serverio konfigūracijų atvejai[2].	11
2 pav.	3 lygių architektūra	12
3 pav.	Žemyn iškrentantis meniu[19]	13
4 pav.	Žemyn iškrentančio meniu punkto pasirinkimas	14
5 pav.	FMS meniu tipas [19].	15
6 pav.	FMS meniu punkto pasirinkimas	15
7 pav.	Ekrano dydžio problema	17
8 pav.	“Top-down” metodo pirmas pavyzdys.	18
9 pav.	“Top-down” metodo antras pavyzdys	19
10 pav.	Mygtuko paspaudimo ir atleidimo problemos. (1) - teisingas mygtuko atleidimas, (2) – klaidingas mygtuko atleidimas	21
11 pav.	Microsoft .NET sudėtis [16]	23
12 pav.	XML žiniatinklio paslaugos suteikia naujas paskirstytųjų komponentų realizavimo galimybes panaudojant standartinius protokolus [16]	24
13 pav.	Vardų sritys, palaikomos .NET Framework ir .NET Compact Framework [16].	26
14 pav.	„TurboStats for Basketball“ [15] ir „PocketStats for Basketball“ [13].	32
15 pav.	„Basketball Statistics Generator“	33
16 pav.	„Basketball Coach“	34
17 pav.	Sistemos išskaidymas į paketus	38
18 pav.	Paketo Prisijungimas Pocket klasių diagrama.	39
19 pav.	Paketo Registracija klasių diagrama	39
20 pav.	Krepšinio varžybų įvykių registravimo sekų diagrama	40
21 pav.	Krepšinio varžybų įvykių registravimo bendradarbiavimo diagrama.	41
22 pav.	Krepšinio varžybų įvykių šalinimo sekų diagrama	41
23 pav.	Krepšinio varžybų įvykių šalinimo bendradarbiavimo diagrama	42
24 pav.	Prisijungimas Pocket klasės būsenų kaitos diagrama	42
25 pav.	Duomenų bazės modelis	43
26 pav.	Išdėstymo diagrama su komponentais	44
27 pav.	Vartotojo sąsajos su 12x12 ir 16x16 taškelių dydžių mygtukais	46

28 pav.	Vidutinių įvykio registravimo laikų palyginimas	47
29 pav.	Santykinių greičių palyginimo diagrama naudojant garsinius pranešimus ir be jų	48
30 pav.	Klaidingų duomenų įvedimo tikimybė.....	48
31 pav.	Vartotojo sąsaja su 20x20 mygtukų dydžiais	55
32 pav.	Vartotojo sąsaja prieš eksperimentą.....	55

Lentelių sąrašas

1 lentelė.	Delninių kompiuterių techninių savybių aprašymas[7].	10
2 lentelė.	Užduoties atlikimo laikų palyginimas naudojant FMS ir „drop-down“ meniu [19] .	16
3 lentelė.	Šešios pažymėjimo strategijos pagal Rein [9].	20
4 lentelė.	Stipriosios ir silpnosios Windows CE .NET programavimo vartotojo sąsajos[16]...	30
5 lentelė.	Alternatyvių sistemų palyginimas	34
6 lentelė.	Vaizdai ir Modeliai	36
7 lentelė.	Eksperimente naudojami mygtukų dydžiai.....	45
8 lentelė.	Įvykio registravimo laikų palyginimas	47
9 lentelė.	Eksperimento rezultatai	56

1. ĮVADAS

Paskutiniu metu, sparčiai plintant interneto paslaugoms, vis dažniau naudojami mobilūs įrenginiai versle, sporte ir buityje. Pasaulyje yra sukurta nemažai programinių sprendimų krepšinio varžybų rezultatų registravimui ir jų analizei. Sukurta programinė įranga labai nesiskyrė viena nuo kitos. Dalis jų turėjo mažiau siūlomų savybių, kita dalis – daugiau. Tačiau pritaikytų mobiliems įrenginiams yra nedaug. Sistema sukurta atsižvelgiant į mobilumo poreikį, rungtynių metu retai yra galimybių pasinaudoti stacionariuoju ar nešiojamu kompiuteriu. Mobilus sistema suteikia laisvę vartotojui – delninis kompiuteris yra patogi priemonė duomenų registravimo ir rezultatų peržiūrėjimo.

Darbo tikslas sukurti mobilią krepšinio varžybų įvykių registravimo ir analizės sistemą, kuri padės vartotojui lengvai ir greitai registruoti krepšinio varžybų įvykius. Užregistruoti įvykiai perkeliama į centrinį serverį, kur atliekama jų analizė, šių dalių projektavimą ir kūrimą atlieka kiti magistrantai.

Projektuojant galima parinkti tokius sprendimus, kurie palengvintų PĮ palaikymą ar pakeitimus, bet reikalautų kompiuterio resursų. Tačiau tų pačių sprendimų pritaikymas delninių kompiuterių programinei įrangai ne visada galimas. Didžiausias dėmesys skiriamas vykdymo greičiui, atminties naudojimui ir ekrano dydžiui[4, 10]. Projektuojant delninių kompiuterių PĮ taip pat neišvengiama problemų. Sprendžiant iškilusias problemas ne visada tinka tie patys sprendimai, kurie tinka asmeninio kompiuterio PĮ kūrimo. Pagrindinės problemos išskylančios projektuojant PĮ skirtą delniniams kompiuteriams:

- vartotojo sąsajos problemos;
- ribotas užduočių atlikimo greitis;
- kintančios aplinkos problema.

Darbe analizuojamos delninių kompiuterių programinės įrangos projektavimo problemos, pateiktos rekomendacijos šioms problemoms spręsti.

Analizuojamos ir palyginamos delninių ir asmeninių kompiuterių egzistuojančios krepšinio varžybų įvykių registravimo ir analizės sistemos. Palyginamos programavimo technologijos, kurios gali būti naudojamos sistemai kurti.

Sukurtos programų sistemos įvertinimui pasirinktas ISO/IEC 9126 kokybės modelis. Šis modelis nekelia reikalavimų programinei įrangai, tačiau gali būti taikomas įvertinti, bet kokios programinės įrangos kokybei. ISO/IEC 9126 standartas pateikia kokybės vertinimo charakteristikas, tačiau nenurodo kokiomis priemonėmis turi būti atliktas vertinimas.

Šiame darbe taip pat atliktas eksperimentinis duomenų įvedimo efektyvumo tyrimas. Modifikuotas sistemos kodas taip, kad galima būtų išmatuoti laikines įvykių registravimo trukmes.

2. ANALIZĖ

2.1 GALIMI SPRENDIMAI

2.1.1 RIBOTAS UŽDUOČIŲ ATLIKIMO GREITIS

Krepšinio rungtynių rezultatų registravimas yra labai intensyvus procesas: reikia spėti užregistruoti labai daug įvykių. Registracija „natūraliu“ būdu (popieriaus lapas ir pieštukas) reikalauja išankstinio pasirengimo: statistinių lentelių sudarymas. Statistinių lentelių pildymas yra varginantis procesas ir rezultatas nėra labai informatyvus. Problemos iškyla, kai naudojamas analizės metodas reikalauja tam tikrų rezultatų, kurie nebuvo surinkti registravimo metu. Kuriama programinė įranga yra specifinio pobūdžio ir nėra orientuota į platų vartotojų ratą. Ji yra skirta krepšinio klubams. Ši įranga bus naudojama kaip trenerio pagalbininkas, leidžiantis registruoti komandos, atskirų žaidėjų veiksmų rezultatus ir atliekantis tam tikrą rezultatų analizę.

Projektuojant mobilią registravimo ir analizės sistemą reikia priimti tokius sprendimus, kad vartotojas galėtų greitai ir efektyviai dirbti. Tačiau tų pačių sprendimų pritaikymas delninių kompiuterių programinei įrangai, kuriuos taikome asmeninio kompiuterio programinei įrangai ne visada galimas.

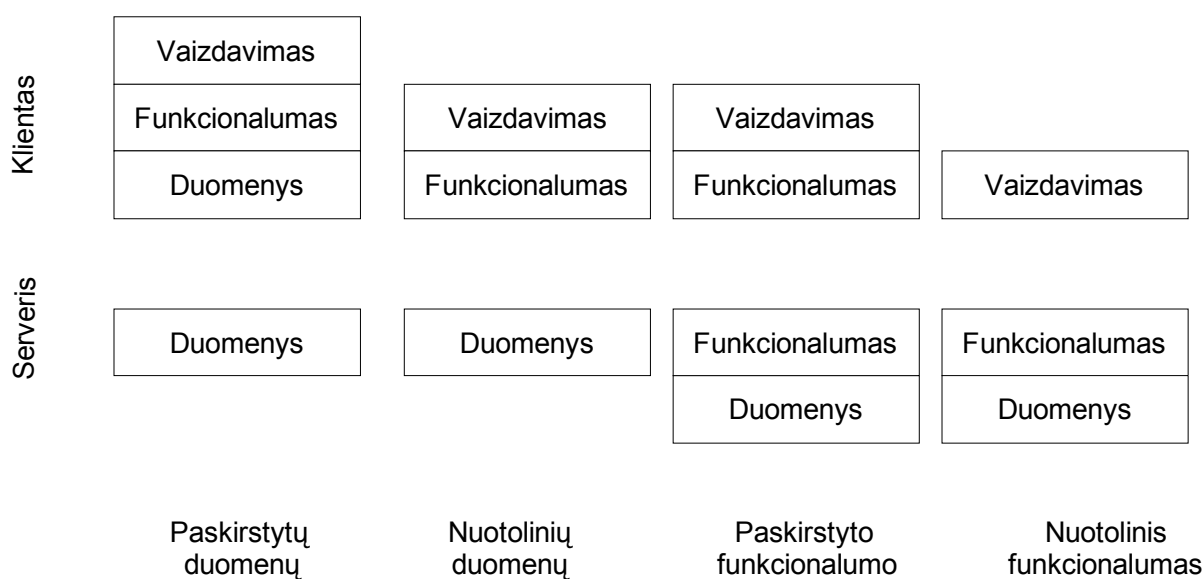
Keletas delninių kompiuterių pagrindinių techninių savybių aprašymai išvardyti 1 lentelėje. Procesoriaus greitis ir atminties dydis kelis ar keliasdešimt kartų mažesnis nei personalinių kompiuterių. Todėl didžiausias dėmesys skiriamas vykdymo greičiui, atminties naudojimui ir ekrano dydžiui[4]. Vartotojo sąsaja turi naudoti kuo mažiau PDA resursų, vykdant pagrindines ir daug resursų reikalaujančias užduotis [4, 6, 17].

1 lentelė. Delninių kompiuterių techninių savybių aprašymas[7]

Modelis	HP IPaq Pocket PC H3950	Asus MyPal A716	Toshiba Pocket PC E405	Toshiba Pocket PC E800
Ekranas dydis	240x320 taškai	240x320 taškai	240x320 taškai	240x320 taškai
Atminties dydis	64MB	64MB	64MB	128MB
Procesorius	300-MHz Intel PXA250	400-MHz XScale	300-MHz XScale	400-MHz XScale

2.1.1.1 KLIENTO SERVERIO KONFIGŪRACIJOS

Projektuojant delninių kompiuterių PĮ svarbu pasirinkti tinkamą kliento ir serverio konfigūraciją. Pasirenkant konfigūraciją pirmiausia yra išskiriami trys komponentai arba lygiai : vaizdavimas, funkcionalumas, duomenys. Visi šie lygiai gali būti įdiegti tiek kliento tiek serverio pusėje. Skirtingos komponentų kombinacijos kliento ar serverio pusėje vadinamos kliento-serverio konfigūracija. Keletas dažniausiai sutinkamų konfigūracijų pateiktos 1 pav.



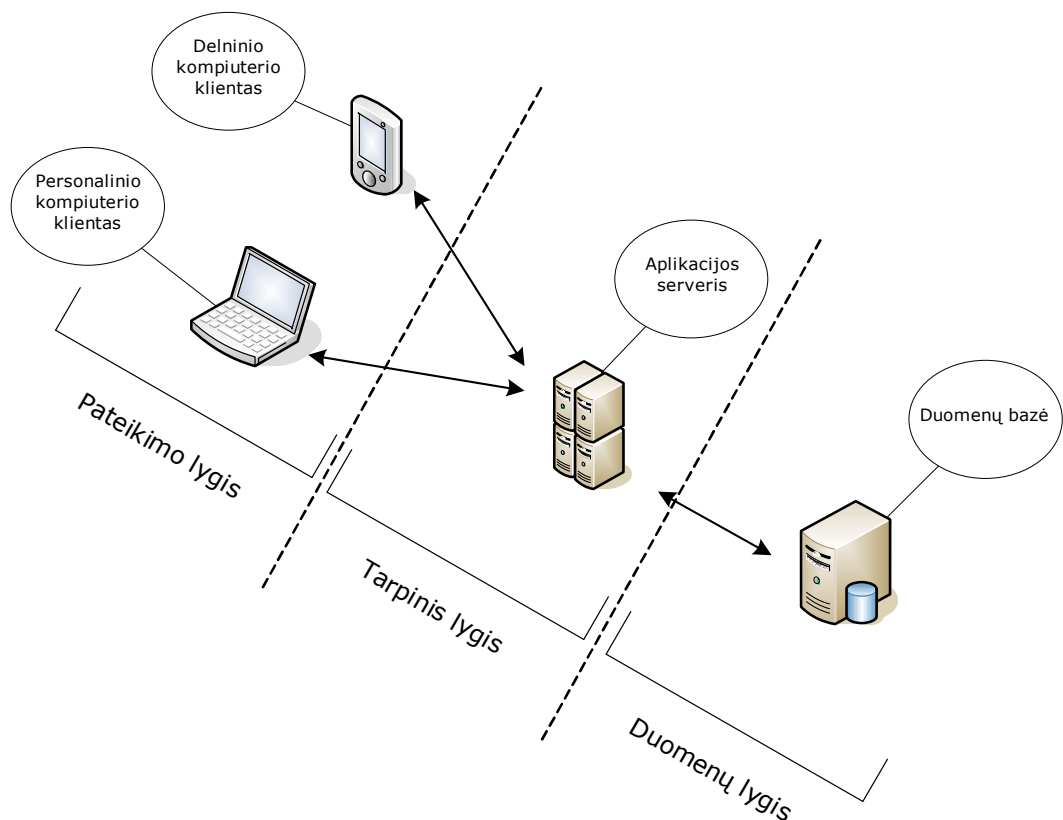
1 pav. Kliento – serverio konfigūracijų atvejai[2]

Kiekvienas lygis yra parenkamas priklausomai nuo tinklo tipo, naudojimo ir kliento techninių galimybių.

- Paskirstytų duomenų – konfigūracija leidžia dirbti neturint pastovaus ryšio su centrine duomenų baze. Tai leidžia lengvai keisti kliento fizinę vietą [2]. Tačiau tokiai konfigūracijai reikalingi nemaži techniniai reikalavimai kliento pusei.

- Nuotolinių duomenų – duomenų bazės serveris yra atskirtas nuo funkcionalumo ir vaizdavimo dalies. Ši konfigūracija leidžia su tais pačiais duomenimis dirbti daug klientų vienu metu, bet tam reikalingas pastovus ryšis su centrine duomenų baze [1, 2].
- Paskirstyto funkcionalumo – konfigūracija leidžia dalį funkcionalumo perkelti į serverio pusę, o kita atlikti kliento pusėje. Naudinga kai tam tikras funkcionalumas gali būti atliekamas tik vienoje platformoje.
- Nuotolinis funkcionalumas - visa funkcionalumas ir duomenys yra centrinio kompiuterio dalyje, kliento pusėje tik vaizdavimas. Dažniausiai sutinkama konfigūracija interneto projektuose [2], kur kliento pusėje yra tikrai naršyklė. Nereikalauja daug resursų kliento pusėje, bet reikalingi resursai serveriui.

Kliento-serverio konfigūracijų yra ir daugiau, šiuo metu dažnai sutinkama 3 lygių architektūra (angl. *3-tier*). Architektūrą sudaro lygiai: vaizdavimo lygis, tarpinis lygmuo (kliento-serverio) konfigūracija ir duomenų lygmuo. 2 paveiksle yra pateikiama sistemos architektūra.



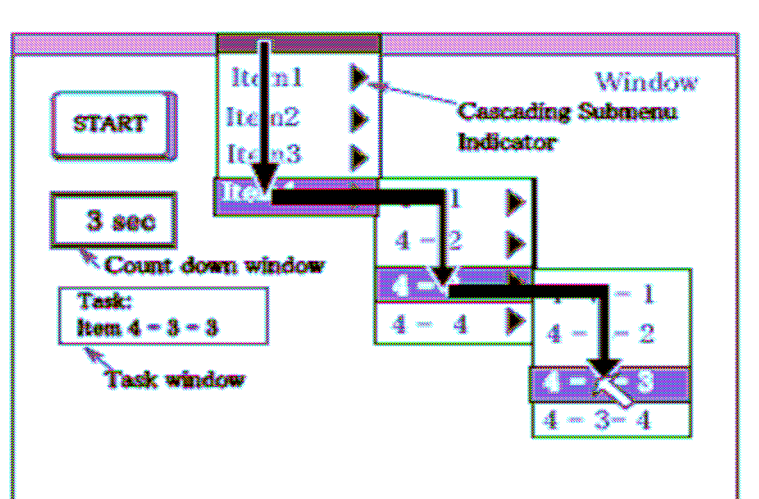
2 pav. 3 lygių architektūra

- *Duomenų* lygyje yra programinės įrangos saugykla.
- *Tarpiniame* lygyje yra realizuojamas programinės įrangos pagrindinis funkcionalumas: duomenų surinkimo valdymas, duomenų išgavimo valdymas, statistikos generavimas, ataskaitų sudarymas.
- *Pateikimo* lygmuo atitinka vartotojo sąsają. Pagrindinis funkcionalumas yra realizuotas tarpiniame lygmenyje.

Tarpinis ir pateikimo lygiai gali turėti ir kitokias konfigūracijas priklausimai nuo taikymo ar reikalavimų.

2.1.1.2 MENU STRUKTŪROS

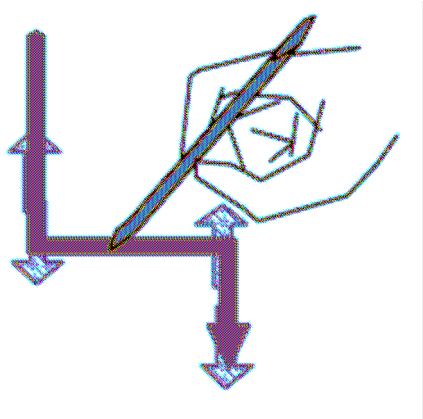
Norint sumažinti užduočių atlikimo laiką reikia sumažinti ir vartotojo atliekamų operacijų laiką. Programose dažnai naudojamas žemyn iškrentančio meniu (angl. *drop-down*) meniu tipas pavaizduotas 3 paveiksliuke.



3 pav. Žemyn iškrentantis meniu[19]

Vartotojui norint pasirinkti meniu punktą esantį antrame lygyje reikia vesti pelės žymeklį du

kartus vertikaliai ir viena kartą horizontaliai 4 paveiksliuke.



4 pav. Žemyn iškrentančio meniu punkto pasirinkimas

Menu punkto pasirinkimo laiką galima paskaičiuoti $\log_2\left(\frac{2*W}{S}\right)$, kur S - meniu punktų kiekis ir

W - atstumas nuo dabartinės žymeklio padėties iki norimo meniu punkto [19].

Reikalingas vartotojo didelis dėmesys norint greitai ir teisingai pasirinkti norimą meniu punktą [19]. Pagyvenusiems ar akliems vartotojams sistemos turinčios žemyn iškrentantį meniu yra sunkiau naudoti[19].

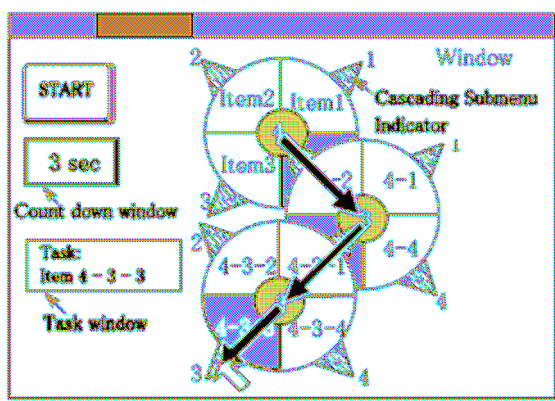
Privalumai:

- galimas didesnis meniu punktų kiekviename lygyje;
- paprastesnis pritaikymas;
- paprasta struktūra;
- sukurta daug komponentų.

Trūkumai:

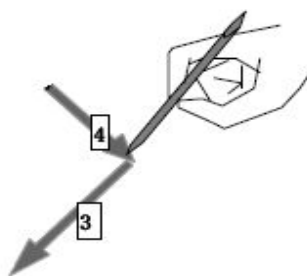
- reikalingas didesnis dėmesys;
- didelė judėjimo amplitudė;
- reikia daugiau laiko išrenkant norimą punktą [17].

FMS (angl. *Float Menu System*) - naujo tipo meniu sistema. Meniu punktai yra išdėstyti ratu pavaizduota 5 paveiksliuke.



5 pav. FMS meniu tipas [19]

Šio meniu tipo naudojimas sistemose leidžia greitai pasiekti norimą meniu punktą, nes reikalinga mažesnė judėjimo amplitudė [19]. Viename lygyje galimi tik keturi meniu punktai, todėl atsiradus daugiau meniu punktų reikalingas papildomas lygis. Vartotojui norint pasirinkti meniu punktą esantį antrame lygyje reikia vesti pelės žymeklį du kartus 6 paveiksliuke



6 pav. FMS meniu punkto pasirinkimas

Privalumai:

- mažesnė judėjimo amplitudė;
- lengvesnis naudojimas.

Trūkumai:

- ribotas galimų punktų kiekis lygyje;
- mažai sukurta komponentų;
- ribotas pritaikymas.

2 lentelė. Užduoties atlikimo laikų palyginimas naudojant FMS ir „drop-down“ meniu [19]

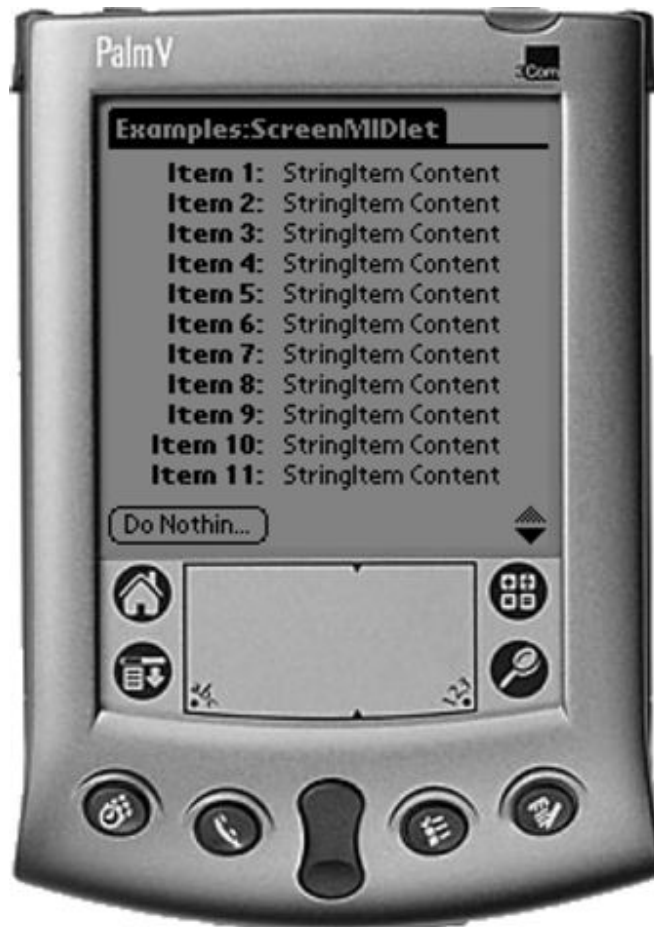
	FMS	„Drop-down“
Suaugusių grupė	4,67 s.	8,45 s.
Jaunimo grupė	2,51 s.	5,00 s.

2.1.2 VARTOTOJO SĄSAJOS PROBLEAMOS

2.1.2.1 RIBOTAS EKRANO DYDIS

Projektuojant svarbus aspektas yra programos paprastas ir lengvas naudojimas. Delninio kompiuterio ekrano dydžiai pateikti 1 lentelėje. PDA įrenginio ekranas kelis kartus mažesnis už asmeninio kompiuterio. Didelis kiekis informacijos nedideliame ekrane, gali pareikalauti didesnio laiko visa informaciją įsisavinti vartotojui. Didėlis skaičius mygtukų ekrane didina mokymąsi dirbti programa laiką [4].

Formos dydis nėra apribotas ir pageidaujant ar pridedant papildomą elementą yra padidinimas. Pvz.: Norite sutalpinti 20 eilutės tipo elementų formoje 7 paveiksle. Šis veiksmas nesukelia problemų, tačiau tik 11 elementų yra matomi ekrane.



7 pav. Ekranu dydžio problema

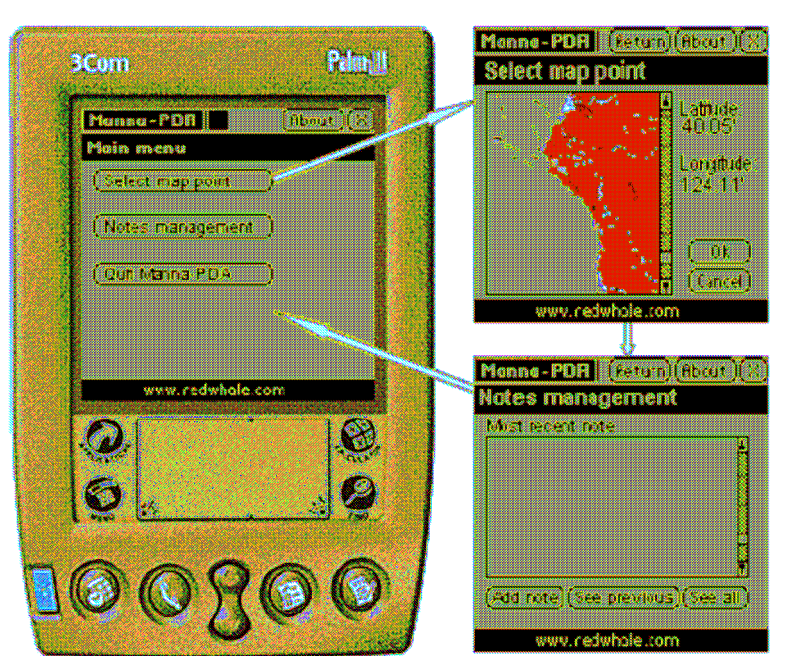
Programa yra draugiškesnė vartotojui jei yra išvengiama poslinkio juostos panaudojimo [4]. Norint išvengti susidariusios problemos rekomenduojama panaudoti sąrašą ar komponentą pasižymintį sąrašo savybėmis.

Įrenginiai turintys mažus ekranus, kur visa informacija negali būti parodyta vienu metu ekrane. Siūloma naudoti šiuos informacijos pateikimo metodus.

„*Top-Down*“ informacijos pateikimo metodas. Šis informacijos pateikimo metodas suteikia informacijos patogesnę pateikimą, sumažina perteklinės informacijos kiekį. Siūlomas informacijos pateikimas daugeliu lygių arba hierarchijos mechanizmo panaudojimas [14, 17].

Pvz.: Vartotojas gauna pranešimą ir jis jo nenori peržiūrėti tiesiog pageidauja atlikti kitą operaciją. Taigi vartotojui nereikia peržiūrėti pranešimo, jis gali nuspręsti tolimesnę seką veiksmų peržiūrėjus tik dalį pranešimo. O norint peržiūrėti visa pranešima tereikia, pasirinkti peržiūrėjimo komandą.

Kitas „*Top-down*“ vartotojo sąsajos pavyzdys pavaizduotas 8 paveiksliuke.



8 pav. „*Top-down*“ metodo pirmas pavyzdys

Siūloma vartotojo sąsają skaidyti į tiek dalių, kad būtų galima pavaizduoti ekrane 9 paveiksliuke.



9 pav. "Top-down" metodo antras pavyzdys

Privalumai:

- informacija lengvai pavaizduojama, bet kokio dydžio ekrane;
- patogesnis informacijos pateikimas [3, 17];
- sumažina perteklinės informacijos kiekį.

Trūkumai:

- didelis formų kiekis;
- reikalingas papildomas funkcionalumas.

2.1.2.2 VARTOTOJO SĄSAJA

Pažymėjimo strategijos

Skirtingos mygtuko pažymėjimo strategijos išvardytos 3 lentelėje.

3 lentelė. Šešios pažymėjimo strategijos pagal Rein [9].

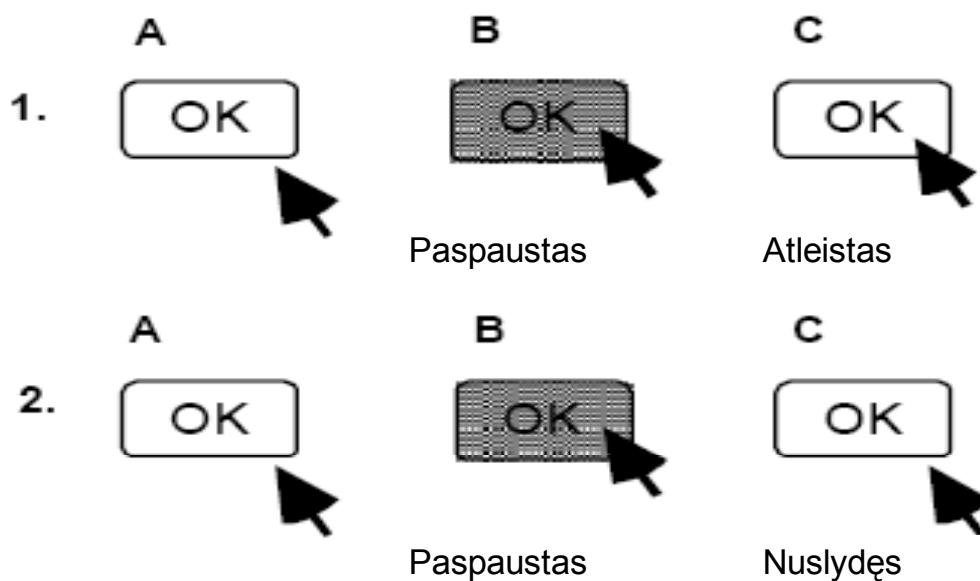
Strategija	Aprašymas
<i>Paspaudus 1</i> (<i>angl. Land-on 1</i>)	Paspaudus mygtuką, atliekami veiksmai.
<i>Paspaudus 2</i> (<i>angl. Land-on 2</i>)	Ši strategija labai panaši į <i>land-on 1</i> . Veiksmai atliekami paspaudus, bet kurioje ekrano vietoje ir užvedus žymeklį ant mygtuko.
<i>Atleidus 1</i> (<i>angl. take-off 1</i>)	Mygtuką paspaudus ir atleidus yra atliekami veiksmai.
<i>Atleidus 2</i> (<i>angl. take-off 2</i>)	Mygtuką paspaudus ir atleidus arba nuvedus nuo mygtuko į šalį yra atliekami veiksmai.
<i>Space 1</i>	Mygtukas pažymimas kai atstumas tarp žymeklio ir mygtuko yra 1 cm. Veiksmai atliekami paspaudus mygtuką.
<i>Space 2</i>	Mygtukas pažymimas kai atstumas tarp žymeklio ir ekrano yra 1 cm.. Veiksmai atliekami paspaudus mygtuką.

Ren, X. ir Moriya, S. atliko eksperimentą, kur buvo tiriama, kuri strategija leidžia greičiausiai pažymėti ir paspausti mygtuką. Eksperimento metu mygtukas pasirodydavo atsitiktinėse ekrano vietose. Geriausias greičio ir klaidų santykis buvo *Land-on 2* strategijos [9]. Realiose situacijose ši strategija turi trūkumą, kai yra daugiau nei vienas mygtukas. Tokiu atveju pažymėjus ekrane ir vedant žymeklį ant norimo mygtuko reikia vengti kitų mygtukų, nes galima paspausti nepageidaujamą mygtuką.

Šiuo metu ne visos strategijos yra diegiamas delniniuose kompiuteriuose. *Space 1* ir *Space 2* strategijos yra taikomos, kur naudojamos elektromagnetinės lentelės (*angl. electromagnetic tablet*).

Informatyvūs atsakomieji pranešimai

Populiariausios strategijos *Land-on1* ir *Take-off 1*, kurios yra naudojamos delniniuose kompiuteriuose [5, 9]. Šių strategijų trūkumas, kad paspaudus mygtuką vartotojui sunku nuspręsti ar mygtukas buvo paspaustas ar ne. Vartotojo sąsaja lengviau naudotis, kai po mygtuko paspaudimo, naudojami atsakomieji pranešimai [9,17]. Mygtuko pažymėjimas kita spalva jo paspaudimo atveju, ne visada padeda išvengti klaidų. Vartotojui gali nutikti tokia situacija, kai mygtukas pažymėtas ir paspaustas teisingai, bet atleidžiant žymeklis nuslydo nuo mygtuko. Situacija pavaizduota 10 paveiksluke. Tokiu atveju veiksmai nebus atliekami, jei naudojama *Take-off 1* strategija. Abiejų atvejų pradžios vienodos 1A ir 2A užvedama ant mygtuko, 1B ir 2B mygtukas paspaudžiamas arba pažymimas, kur mygtuko spalva pasikeičia. Tačiau 1C ir 2C atvejai skiriasi, nors mygtuko spalva pasikeitė. Pirmu atveju mygtuko paspaudimas yra teisingas ir veiksmai yra atliekami, o antru atveju mygtuko veiksmai klaidingi, bet sistema vartotojui nepraneša klaidingo mygtuko paspaudimo atvejo.



10 pav. Mygtuko paspaudimo ir atleidimo problemos. (1) - teisingas mygtuko atleidimas, (2) – klaidingas mygtuko atleidimas

Sistema turėtų pranešti apie atliekama operaciją, kaip paspausta klavišą trumpu pyptelėjimu ar klaidos pranešimą įvedant klaidingus duomenis [17]. Pranešimai turi būti aiškūs ir suprantami vartotojui. Pvz.: pranešimas „HTTP404 ERROR“ ir „THE PAGE CAN NOT BE FOUND“ gali būti ekvivalentūs, bet daugumai vartotojų gali būti labai naudinga informacija.

2.1.3 KINTANČIOS APLINKOS PROBLEMA

Rekomendacijos kintančios aplinkos problemai spręsti:

- galimybė keisti išvedamų duomenų savybes: teksto dydį, šviesumą;
- leisti atlikti operacijas viena ranka arba balsu;
- galimybė dirbti programa neturint interneto ryšio;
- programa turi galimybę prisitaikyti prie vartotojo nustatytos aplinkos.

Programų skirtų PC ir naudojamų ofise ar namuose nustatymai yra panašūs. Mobilių įrenginių PĮ nustatymai priklauso nuo aplinkos. Aplinkos pasikeitimai (šviesumas, triukšmingumas, oras) gali pasikeisti priklausomai nuo vietos, laiko ar sezono. Programos naudojimą ar tinkamumą gali pakeisti įvairūs aplinkos faktoriai[17]. Taigi aplinkos pasikeitimas pareikalauti pakeisti programos nustatymus Pvz.: Įvedimas balsu vartotojui gali būti nepatogus ar nemalonus. Pvz.: bibliotekoje ar tarp nepažystamų žmonių. Taigi reikalinga galimybė atsisakyti įvedimo balsu ir pasirinkti kita alternatyvą.

Išvedamo teksto dydis tinka dirbti ofise, bet staiga pasikeitus aplinkai ar apšvietimui jau tas pats tekstas gali būti neįskaitomas.

Aplinkos pasikeitimo problemos sprendimui reikalinga įdiegti „*context-awareness*“ ir prisitaikymo funkcionalumą [18]. Šios problemos išsprendimas gali padidinti programos panaudojimą ir išsaugoti vartotojo pasitikėjimą [17]. Programos panaudojimas dinamiškoje aplinkoje gali būti išspręstas panaudojant duomenų įvedimui įrenginį, kuris gali matyti, girdėti ir jausti (angl. *see, hear, and feel*) .

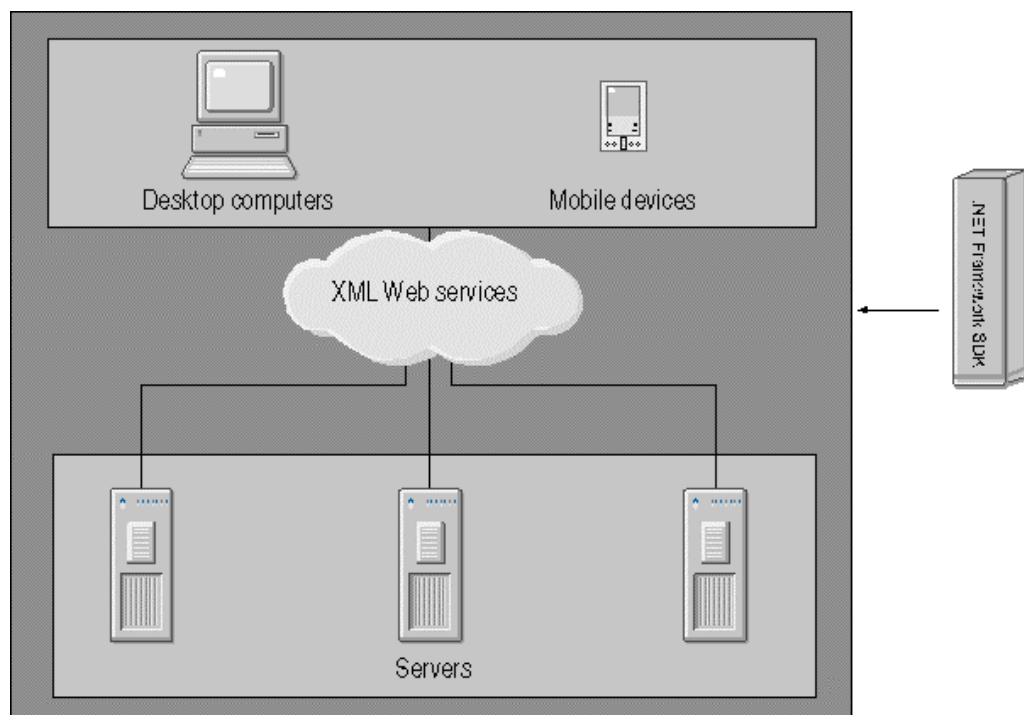
2.2 TECHNOLOGIJŲ ANALIZĖ

2.2.1 MICROSOFT .NET TECHNOLOGIJOS

Pagrindinės kūrimo priemonės yra .NET technologijos, kurios susideda iš labai plataus spektro ir įvairaus masto projektavimo technologijų, todėl buvo atliekama tik sekančių technologijų analizė:

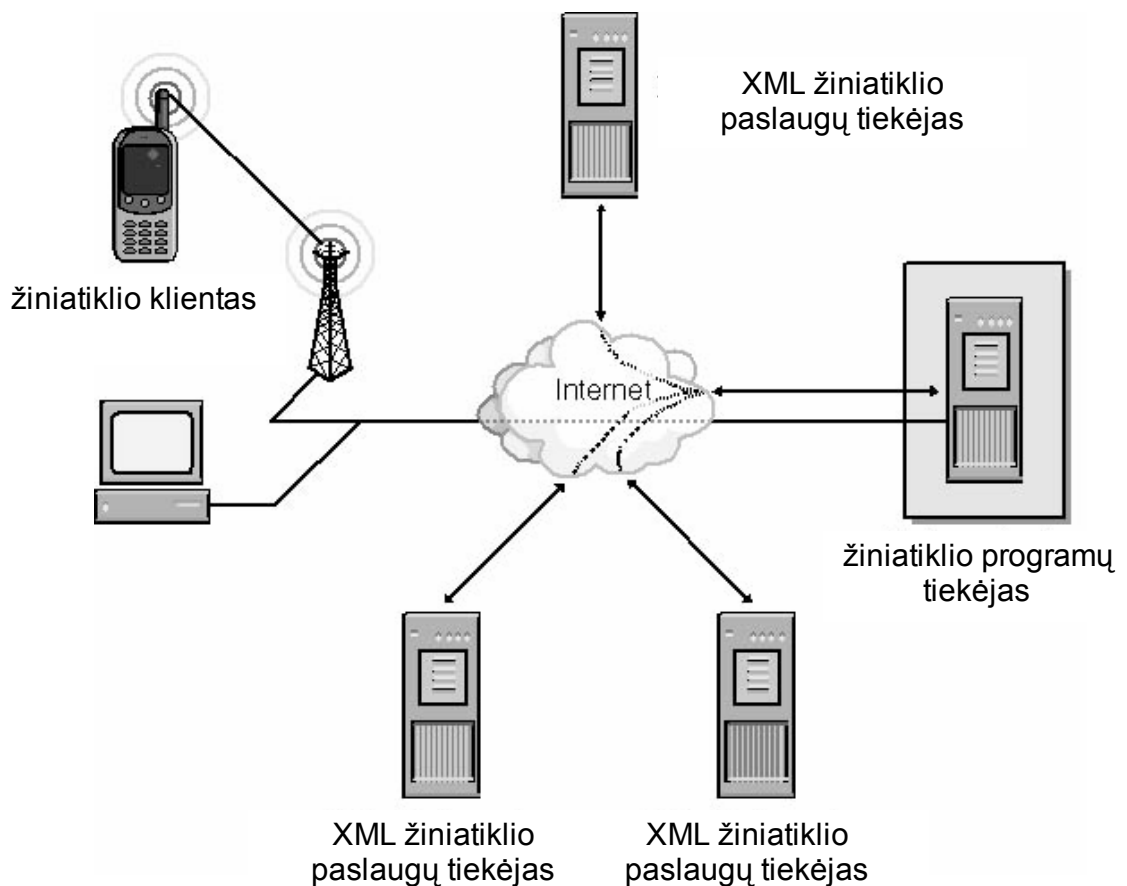
- *.NET Framework*
- *.NET Compact Framework*
- *Visual Studio*

Microsoft .NET yra programinė įranga kuri apima informaciją, žmones, sistemas ir įrenginius. Ji apima klientus, serverius, kūrimo įrankius ir susideda iš tokių dalių:



11 pav. Microsoft .NET sudėtis [16]

.NET Framework 1.1 naudojamas ne tik programinei įrangai kurti, bet ir su ja dirbti. Įtraukiant interneto pagrindu sukurtas programas, intelektualiujų klientų programas ir XML žiniatiklio paslaugas – komponentai kurie palengvina apsikeitimo duomenimis funkcionalumą naudojant standartinius nuo platformos nepriklausomus protokolus, kaip XML, SOAP ir HTTP.



12 pav. XML žiniatiklio paslaugos suteikia naujas paskirstytųjų komponentų realizavimo galimybes panaudojant standartinius protokolus [16]

- Tokių kūrimo įrankių, tokių kaip *Microsoft Visual Studio® .NET 2003*, kuriame integruota kūrimo aplinka (IDE) maksimizuoja PĮ kūrėjo produktyvumą dirbant su *.NET Framework*.

- Aibės serverių, įtraukiant *Microsoft Windows® Server 2003*, *Microsoft SQL Server™*, ir *Microsoft BizTalk® Server*, kurie integruoja, vykdo, operuoja ir valdo žiniatiklio paslaugas ir interneto pagrindu sukurtas programas.
- Tokios klientinės programinės įrangos, kaip *Windows XP*, *Windows CE*, ir *Microsoft Office XP*.

.NET Framework yra integralus *Windows* komponentas leidžiantis kurti ir dirbti su sekančios kartos programine įranga. *.NET Framework* yra suprojektuotas atlikti šioms užduotims[12]:

- suteikti pastovią objektinę programavimo aplinką;
- suteikti kodo vykdymo aplinką, kuri minimizuotų programinės įrangos įdiegimo ir versijų skirtingumo konfliktus;
- suteikti tokią kodo vykdymo aplinką, kuri palaikytų saugų kodo vykdymą, įtraukiant sukurtą nežinomų ar mažai patikimų trečių šalių kodą;
- užtikrina, kad kodas parašytas *.NET Framework* gali integruotis su bet kuriuo kitu kodu.

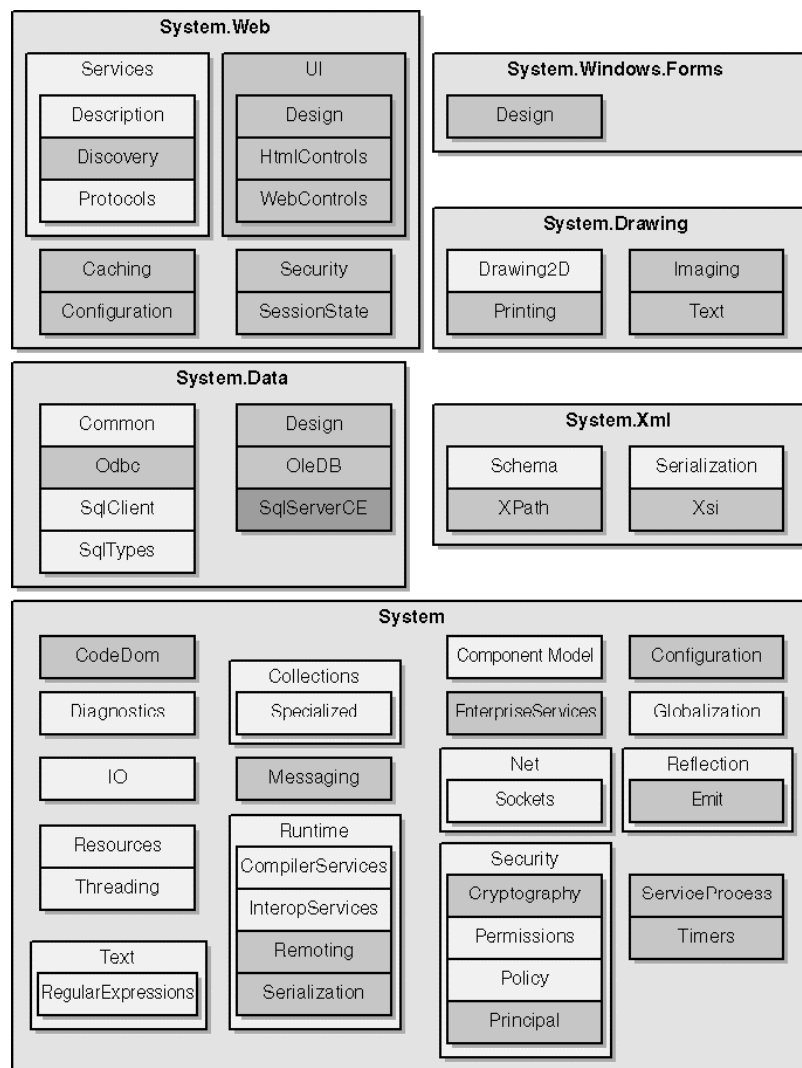
.NET Framework savybės:

- Palaiko daugiau negu 20 skirtingų programavimo kalbų
- Labiau negu kas kitas anksčiau palengvina programinės įrangos kūrimą, diegimą, ir saugumo administravimą sudėtingoms ir didelio našumo reikalaujančioms programoms.

Su *.NET Framework* įmanomas greitas programinės įrangos kūrimas. Daugelio kalbų palaikymo galimybė įgalina PĮ kūrėjus naudoti tą kalbą, kuri labiausia tinka konkrečiai užduočiai atlikti ir kurti tą pačią programinę įrangą naudojant skirtingas programavimo kalbas. Komponentai parašyti skirtingomis programavimo kalbomis gali naudotis vienas kito funkcionalumu be papildomo PĮ kūrėjo įsikišimo.

Microsoft .NET Compact Framework yra intelektualiujų įrenginių kūrimo platforma pirminiam *Microsoft .NET* ir dalis *Microsoft* tikslo pateikti sukurta programinę įrangą – bet kuriuo metu bet kurioje vietoje ir bet kokio tipo įrenginyje. *.NET Compact Framework* susideda iš valdomo kodo ir XML žiniatiklio paslaugos intelektualiams įrenginiams, ir įgalina vykdymą saugių, parsisiunčiamų programų PDA įrenginiams, mobiliems telefonams.

.NET Framework ir *.NET Compact Framework* sudarą galima pavaizduoti grafiškai, nurodant, kurios klasės yra palaikomos vienoje ar kitoje technologijų bazėje. Sandara pavaizduota 13 paveiksle: pilkai išskirtos klasės yra nerealizuotos *.NET Compact Framework* realizacijoje.



13 pav. Vardų sritys, palaikomos *.NET Framework* ir *.NET Compact Framework* [16]

Paveiksle 13 matome išskirtą vardų sritį *System.Data.SqlServerCE*. Ši vardų sritis realizuota *.NET Compact Framework* bazėje ir *.NET Framework* neturi jos atitikmens.

Pagrindiniai *.NET Compact Framework* ir *.NET Framework* skirtumai:

- *.NET Compact Framework* pateikia klases, kurias galima panaudoti infraraudonųjų spindulių ryšio apdorojimui;
- *.NET Compact Framework* pateikia naują API, kurio siunčia ir gauna duomenys iš infraraudonųjų spindulių prievado;
- *.NET Compact Framework* palaiko tik žiniatiklio paslaugų kliento realizaciją;
- *.NET Compact Framework* nepalaiko gijų prioritetų sistemos, kuri egzistuoja *.NET Framework*.

Palyginti *.NET Framework* su *.NET Compact Framework* turi mažesnę klasių bibliotekų rinkinį. Taip pat *.NET Compact Framework* yra optimizuotas atsižvelgiant į dydį ir veikimo laiką. *.NET Framework* vykdomoji dalis yra apie 30 MB, tuo tarpu *.NET Compact Framework* užima apie 1.5 MB.

Kadangi *Microsoft .NET Compact Framework* yra *.NET Framework* poaibis, PĮ kūrėjai turintys *.NET Framework* patirties gali dirbti su *.NET Compact Framework*. *Microsoft* ateityje pristatys praplėtimus *Visual Studio .NET*, pavadintus intelektualių įrenginių programavimo *Visual Studio .NET*, kurių tikslas *.NET Compact Framework*. Kadangi kiekvienas *Visual Studio .NET* programuotojas, taip pat bus ir intelektualiujų įrenginių programinės įrangos programuotojas, todėl eksponentiškai *.NET Compact Framework* programuotojų bus daugiau negu kitų įrenginių arba mobilių programavimo platformų.

.NET Compact Framework privalumai:

- bendras kodas ir padidintas efektyvumas;
- pritaikomas kiekvienam įrenginiui;
- saugiai įvykdomas.

Kadangi *.NET Compact Framework* pristato tą patį programinį modelį tarp įvairių įrenginių, tai supaprastina PĮ kūrimo procesą, kuris bus vykdomas skirtinguose įrenginiuose. Dalis programos, kaip verslo logika, duomenų prieigos sluoksnis ir XML žiniatiklio paslaugų sluoksnis, gali būti padalintas skirtingiems įrenginiams. Tai padidina programinės įrangos kūrimo efektyvumą [8].

.NET Compact Framework yra suprojektuotas taip, kad galėtų būti išplečiamas su klasių bibliotekomis, kurios tiekia unikalias savybes tam tikroms įrenginių šeimoms ar net konkrečiam

įrenginio modeliui. Toks požiūris leidžia PĮ kūrėjams sukurti geriausią PĮ specifiniam įrenginio modeliui, pakartotinai panaudojant įgūdžius ir kodą.

.NET Compact Framework siūlo tvirtą ir saugią aplinką. Valdomo kodo modelis, palaikomas *.NET Compact Framework*, padidina kodo patikimumą, tokiu būdu sumažindamas programinės įrangos defektus. Valdomo kodo vykdymo variklis užtikrina, kad prastai veikianti programa negalėtų sugadinti paties įrenginio. Tuo pačiu metu saugumo modelis įdiegtas į *.NET Compact Framework* užtikrina, kad piktybiniam kodui nebus leidžiama prieiti prie apsaugotų sistemos resursų.

Visual Studio .NET 2003 siūlo mobilių technologijų programuotojams daug galimybių. Tiek kurti mobilias interneto programas, kurių tikslas būtų platus ratas skirtingų mobilių įrenginių, tiek pilnai eksploatuoti intelektualiesiems įrenginiams, kaip *Windows Mobile Pocket PC* ar *Pocket PC 2002*, *Visual Studio .NET 2003* turi visas tam reikalingas galimybes.

Į *Visual Studio .NET 2003* intelektualiujų įrenginių programavimo savybes įeina:

- Microsoft *.NET Compact Framework*;
- papildomi projektų tipai, skirti *Microsoft Visual C# .NET* ir *Microsoft Visual Basic .NET*;
- integruotas Visual Microsoft Windows formų dizaineris;
- nuotolinių įrenginių derinimas;
- intelektualiujų įrenginių emuliacija.

Visual Studio .NET 2003 turi integruotą *.NET Compact Framework* palaikymą. Naudojant *Microsoft Windows* formų dizainerį, PĮ kūrėjai gali kurti, derinti ir diegti programas intelektualiesiems įrenginiams. *.NET Compact Framework* veikia tik su *Windows Mobile PC* ir *Windows CE* operacinėmis sistemomis.

2.2.2 MICROSOFT WINDOWS 32-BIT API GALIMYBĖS

Win32 API suteikia galimybę kurti tokius žemo lygio komponentus, kaip įrenginių tvarkyklės ir dinaminės bibliotekos (angl. *dynamic-link library*), kurios praplečia operacinę sistemą. Bendri Win32 API bruožai, kurie gali būti optimalūs arba vieninteliai sprendimai tam tikrose taikomosiuose programose [16]:

- greičiausi vykdomieji failai;
- geriausias tikralaikio (angl. *real-time*) palaikymas;
- išėties kodo (tarp platformų) mobilumas;

- galimybė kurti įrenginių tvarkykles;
- galimybė kurti valdymo pulto aplikacijas;
- pasirinktinių vartotojo sąsajos apvalkalų palaikymas;
- saugumo išplėtimo palaikymas;
- delninio kompiuterio išplėtimo palaikymas.

Greičiausi vykdomieji failai

Viena iš priežasčių yra ta, kad Win32 vykdomieji failai pateikiami kaip savas mašininis kodas.

Geriausias tikralaikis palaikymas

Win32 geriausiai naudoti tikralaikėms sistemos projektuoti dėl to, kad Win32 vykdomieji failai yra vykdomi greičiausiai.

Išeities kodo (tarp platformų) mobilumas

Win32 ir *.NET Compact Framework* taikomosios programos suteikia tam tikrą mobilumo laipsnį: Win32 suteikia išeities kodo mobilumą, o *.NET Compact Framework* suteikia binarinį mobilumą. Win32 mobilumas pasitarnauja, kai tam tikras kodas turi būti vykdomas ant daugelio *Windows CE* platformų, net ant tų, kurios neturi *.NET Compact Framework* vykdymo aplinkos. Win32 vykdomųjų failų apribojimai: jiems reikalingos atitinkamos Win32 funkcijos ir jie yra priklausomi nuo CPU.

Galimybė kurti įrenginių tvarkykles

Visos tvarkyklės turi būti rašomos naudojant Win32.

Priežastys: užimama vieta, greitas vykdymas, tikralaikis palaikymas ir perkėlimo galimybė ant didesnio kiekio platformų.

Architektūros atžvilgiu įrenginių tvarkyklės yra tam tikro tipo dinaminės bibliotekos, kurios negali būti sukurtos naudojant *.NET Compact Framework*.

Galimybė kurti valdymo pulto aplikacijas

Yra galimybė įdėti naujas piktogramas į *Windows CE* operacinės sistemos valdymo pultą. Tokiu būdu sukuriama centralizuota sritis, kurioje vartotojai gali surasti ir keisti sistemos nustatymus.

Saugumo išplėtimo palaikymas

Galima teigti, kad Win32 vykdomieji failai yra saugesni negu *.Net Compact Framework* vykdomieji failai. Viena iš priežasčių yra ta, kad *s* vykdomieji failai saugo metaduomenis, kurie pateikia nustatymų vardus, metodus ir įvykius.

Delninio kompiuterio išplėtimo palaikymas

Delninio kompiuterio branduolys leidžia atlikti tam tikrus patobulinius, kurie turi būti realizuojami su Win32.

2.2.3 MICROSOFT WINDOWS 32-BIT API IR .NET COMPACT FRAMEWORK PALYGINIMAS

Dviejų technologijų palyginimas turi būti atliekamas apžvelgiant ne tik teigiamus, bet ir neigiamus bruožus. Pagrindiniai Win32 ir *.NET Compact Framework* bruožai pateikti 4 lentelėje.

4 lentelė. *Stipriosios ir silpnosios Windows CE .NET programavimo vartotojo sąsajos*[16]

API	Stiprybės	Silpnybės
<i>Microsoft Win32 (C / C++)</i>	<ul style="list-style-type: none">○ Geriausias tikralaikio palaikymas.○ Išėities kodo (tarp platformų) mobilumas.○ Reikalingas įrenginių tvarkyklėms.○ Reikalingas branduolio išplėtimams.○ Nėra papildomos vykdymo aplinkos.○ Saugumo išplėtimo palaikymas.○ Delninio kompiuterio išplėtimo palaikymas.	<ul style="list-style-type: none">○ Atminties atlaisvinimas nėra automatizuotas.○ Žemo lygio API.○ Ne objektinė, o procedūrinė API.

<p><i>.NET Compact Framework</i></p>	<ul style="list-style-type: none"> ○ Gerai suprojektuota programuotojo sąsaja. ○ Lengvai kuriama vartotojo sąsaja. ○ Objektinė API. ○ Tipų saugumas. ○ Patogi duomenų bei duomenų bazių valdymo realizacija. ○ Automatinis atminties šiukšlių surinkimas ○ Mobilios mašininių komandų aibė. ○ Lengvai ir greitai kuriami žiniatinklio paslaugų klientai. ○ Puikus XML valdymo palaikymas. ○ Puikus kūrimo įrankių palaikymas. ○ Binarinis mobilumas keliems CPU. 	<ul style="list-style-type: none"> ○ Vykdyto aplinkos apimtis. ○ Dideli nuostoliai vykdant kvietimus tarp valdomo ir nevaldomo kodo. ○ COM operacinis suderinamumas šiek tiek griezdiškas.
--------------------------------------	---	---

2.3 PANAŠIŲ SISTEMŲ APŽVALGA

Apžvelgiamos sukurtos delninių kompiuterių sistemos. Jų veikimo principai, techniniai ir architektūriniai sprendimai, kurias galima panaudoti savo kuriamoje sistemoje.

Duomenų įvedimas vykdomas pažymint delninio kompiuterio ekrane. Ekrane yra išpiešta krepšinio aikštelės 2D vaizdas, tai leidžia greitai pažymėti mėtimo ar kito įvykio vietą. Toks būdas yra patogus ir greitas įvesti varžybų duomenis. Vaizdinė informacija vartotojui pateikia daugiau informacijos, o tai leidžia lengviau, greičiau perprasti sistemą vartotojui.

„*TurboStats for Basketball*“ programos pagrindinis langas pavaizduotas 14 paveiksliukas pirmas piešinys. Fiksuoti krepšinio varžybų įvykius naudojama ekrane nubraižytą krepšinio aikštelę. Šios programos pagrindinės savybės:

- duomenų įvedimas ekrane;
- galimybė dirbti neturint interneto ryšio;
- galimybė įvesti balsu;
- tikralaikė duomenų sinchronizacija su asmeniniu kompiuteriu.



14 pav. „TurboStats for Basketball“ [15] ir „PocketStats for Basketball“ [13]

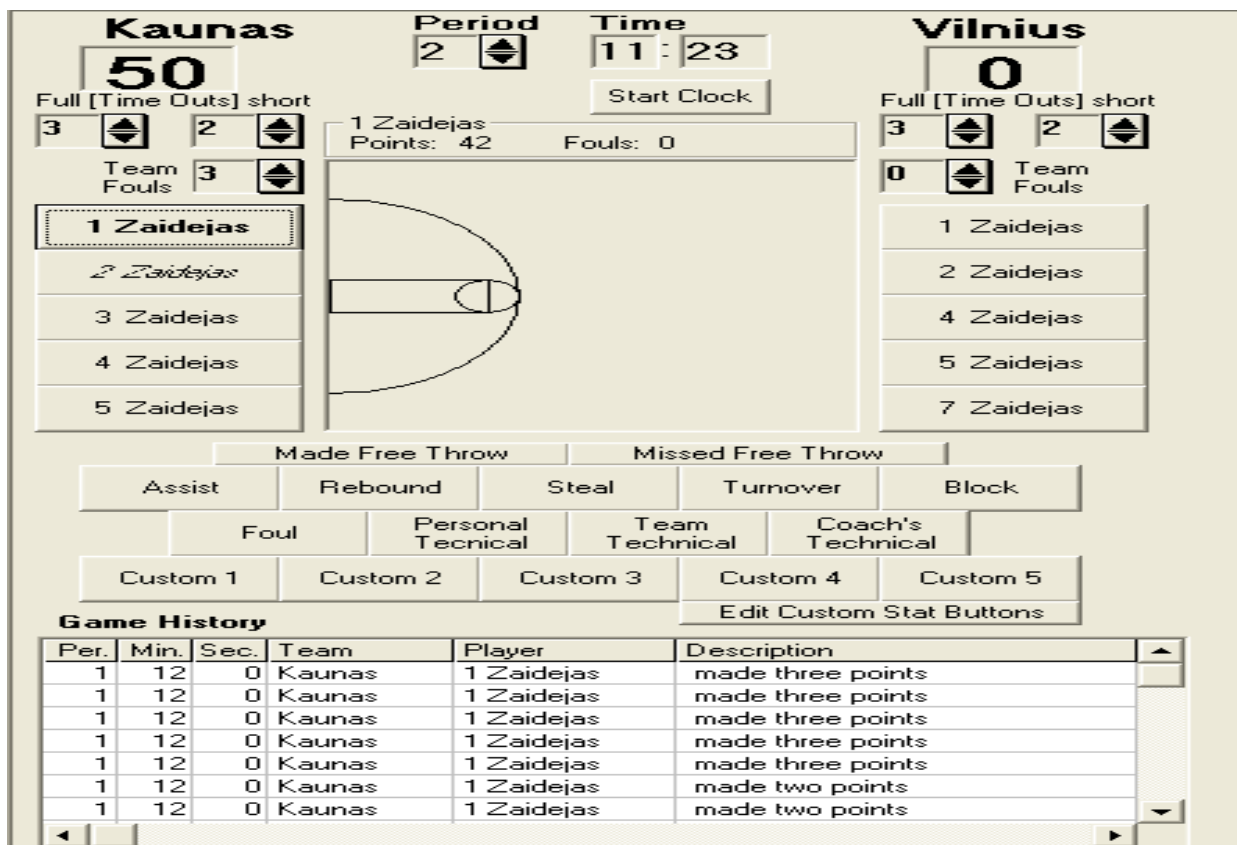
„PocketStats for Basketball“ skirta registruoti krepšinio varžybų įvykių registracijai ir peržiūrai. „PocketStats for Basketball“ ir „TurboStats for Basketball“ programų funkcionalumas labai panašus, tačiau jos turi keletą svarbių skirtumų. Krepšinio varžybų įvykius fiksuoti naudojama ekrane nubraižytą krepšinio aikštelę. Šios programos pagrindinės savybės:

- duomenų įvedimas ekrane;
- galimybė dirbti neturint interneto ryšio;
- galimybė įvesti balsu;
- neturi tikralaikės duomenų sinchronizacijos su asmeniniu kompiuteriu.

„Basketball Statistics Generator“ pavaizduota 12 paveiksluke, taškų pelnyto fiksavimui naudoja grafinį aikštelės vaizdą. Metimų metu yra fiksuojama metimo padėtis bei laikas, tačiau realiame laike metimų rezultatai nėra pateikiami. Tokiu būdu nėra apkraunamas duomenų įvedimo laukas. Metimų rezultatus galima pažiūrėti pasirinkus atitinkamą meniu punktą.

Taip pat yra fiksuojami kiti įvykiai: pražangos, kamuolio praradimai, atkovojojimai ir t.t.

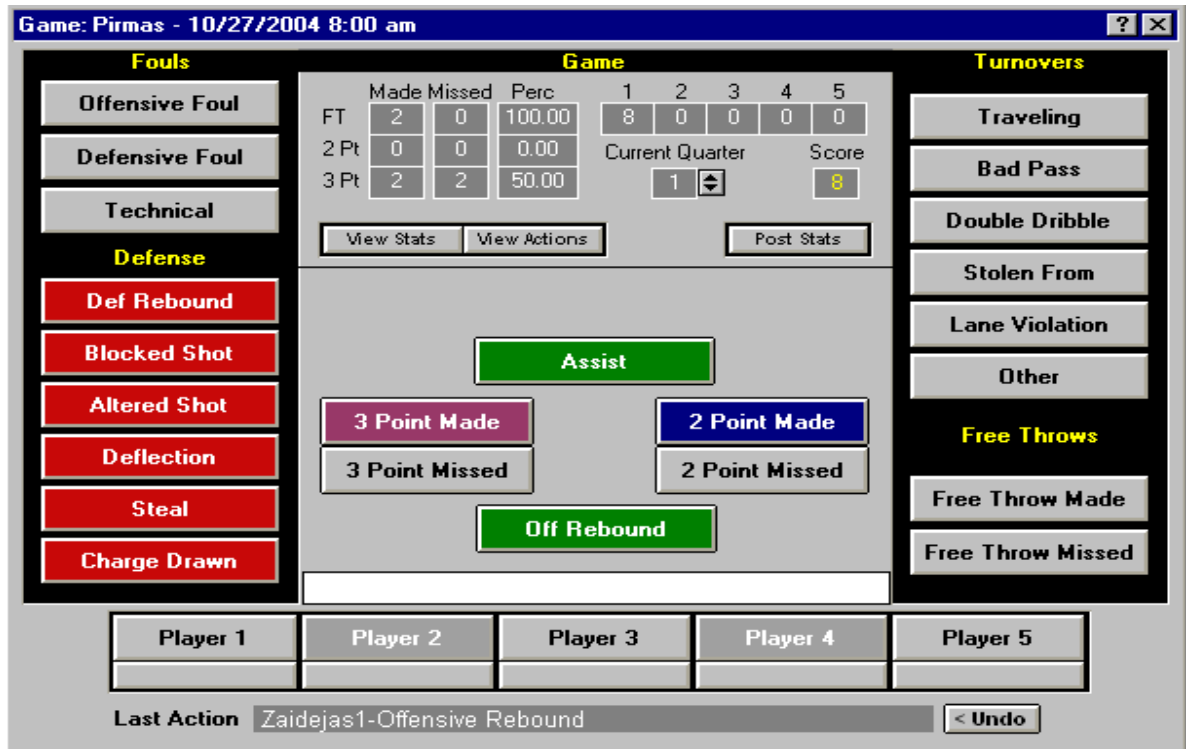
Yra galimybė įsivesti papildomus fiksuojamus įvykius (iki 5 įvykių). Fiksuojamas visų įvykių laikas. Pateikiama visa įvykių istorija, kuri gali būti lengvai koreguojama ir pašalinami klaidingi įvykiai.



15 pav. „Basketball Statistics Generator“

„Basketball Coach“ aikštelė nėra grafiškai pavaizduota vartotojo sąsajoje, o tik parinkti galimos įvykių baigtys, bet tai neleidžia užfiksuoti įvykio vietos pavaizduota 16 paveiksliuke.

Šioje sistemoje yra realizuota galimybė rezultatus įvedinėti balso komandų pagalba. Taipogi gera alternatyva grafiniams įvedimo būdams, kas padeda greičiau įvesti duomenis, bet kartu padaro sistemos realizavimą sudėtingesnį.



16 pav. „Basketball Coach“

5 lentelė. Alternatyvių sistemų palyginimas

PĮ	Veikimas		Iškirtinės savybės
	PC operacinė sistema	PDA operacinė sistema	
„TurboStats for Basketball“	Windows	Palm OS 3.0	Įvedimas balsu, tikralaikė duomenų sinchronizacija
„PocketStats for Basketball“	-	Palm OS 3.1	-
„Basketball Statistics Generator“	Windows	-	Papildomų įvykių galimybė
„Basketball Coach“	Windows	-	Įvedimas balsu

2.4 PRIIMTO SPRENDIMO PAGRINDIMAS

Atlikus analizę buvo nuspręsta sukurti mobilią sistemą, kuri leistų registruoti krepšinio varžybų rungtynių rezultatus ir vėliau juos analizuoti. Kuriama programinė įranga yra specifinio pobūdžio ir nėra orientuota į platų vartotojų ratą. Ji yra skirta krepšinio klubams. Ši įranga bus naudojama kaip trenerio pagalbininkas, leidžiantis registruoti komandos bei atskirų žaidėjų veiksmų rezultatus ir atliekantis, tam tikrą rezultatų analizę.

Sistema turi būti mobili, todėl buvo nuspręsta naudoti delninį kompiuterį, nes tai leidžia vartotojui lengviau naudoti programą įvairiose situacijose, negu naudojant personalinį kompiuterį. Renkantis kliento-serverio konfigūraciją, buvo nuspręsta naudoti dviejų konfigūracijų sujungtą konfigūraciją: paskirstytų duomenų ir paskirstyto funkcionalumo. Paskirstytų duomenų konfigūracija naudojama, nes vartotojo reikalavimas dirbti su sistema neturint interneto ryšio. Paskirstyto funkcionalumo konfigūracija naudojama, nes duomenų analizės funkcionalumas yra vykdomas centriniame kompiuteryje. Naudojant PDA atsiranda kitų problemų, kaip ribotas ekrano dydis. Šiai problemai buvo pasirinkta naudoti *top-down* informacijos pateikimo metodą. Renkantis meniu struktūrą iš *FMS ir Drop-down* nuspręsta naudoti standartinę *drop-down* meniu dėl paprastos vartotojo meniu struktūros ir kad meniu pagalba bus iškviečiamos tik retai naudojamos komandos. Pagrindinių komandų iškvietimui bus naudojami mygtukai su atsakomaisiais garsiniais pranešimais, mygtuko paspaudimui pranešti. Šie atsakomieji pranešimai naudojami, norint išvengti nuslydimos problemos.

Įvykių vietos nustatymui naudojama ekrane yra išpiešta krepšinio aikštelės vaizdas, tai leidžia greitai pažymėti mėtimo ar kito įvykio vietą. Toks būdas yra patogus ir greitas įvesti varžybų duomenis. Vaizdinė informacija vartotojui pateikia daugiau informacijos, o tai leidžia lengviau, greičiau perprasti sistemą vartotojui.

Renkantis kūrimo platformą, svarstyta keletas alternatyvų: *.NET* ir *Win 32* technologijos. Galutinis pasirinkimas buvo *.NET* technologijos dėl plataus kūrimo, palaikymo priemonių, patogios programuotojo sąsajos ir vartotojo reikalavimų sistemai dirbti *Windows* aplinkose.

3. PROJEKTINĖ DALIS

3.1 BENDRA APŽVALGA

Architektūra pateikiama laikantis UML nustatytų standartų, bei atsižvelgiant į RUP rekomendacijas. Naudojami šie pagrindiniai sistemos architektūrą aprašantys vaizdai ir modeliai kuriais remiantis jie buvo projektuojami:

6 lentelė. Vaizdai ir modeliai

VAIZDAI	MODELIAI
Panaudojimo atvejų vaizdas	PA diagrama (angl. <i>CASE diagram</i>)
Sistemos statinis ir dinaminis vaizdas	Klasių diagrama (angl. <i>Class diagram</i>) Veiklos diagrama (angl. <i>Activity diagram</i>) Sekų diagrama (angl. <i>Sequence diagram</i>) Bendradarbiavimo diagrama (angl. <i>Collaboration diagram</i>) Būsenų diagrama (angl. <i>State diagram</i>)
Išdėstymo vaizdas	Komponentų diagrama (angl. <i>Component diagram</i>) Išdėstymo diagrama (angl. <i>Deployment diagram</i>)
Duomenų vaizdas	Duomenų bazės modelis (angl. <i>Database diagram</i>)

3.2 ARCHITEKTŪRINIS VAIZDAS

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

Panaudojimo atvejų vaizde sudaroma panaudojimo atvejų diagrama bei pateikiamos pagrindinių panaudojimo atvejų specifikacijos. Šis vaizdas įtakoja kituose vaizduose pateiktas diagramas.

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

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

Paskirstymo vaizde parodoma, kokie aparatūriniai komponentai sudarys kuriamą sistemą.

3.3 ARCHITEKTŪROS TIKSLAI IR APRIBOJIMAI

Architektūriniai tikslai:

Panaudoti tokią architektūrą, kuri leistų duomenis surinkinėti delniniame kompiuteryje, bei juos ten atvaizduoti, apdoroti, kausti.

Turi leisti surinkinėti rungtynių rezultatus keliuose delniniuose kompiuteriuose vienu metu, bei sukauptus duomenis apjunktį. Atsiradus ryšiui su centrine duomenų baze, juos perkelti į centrinę duomenų bazę arba iš jos pasiimti reikiamus duomenis.

Duomenų įvedimas turi būti atliekamas tiesiai ekrane pasinaudojant vartotojo grafine sąsaja.

Duomenų apsikeitimui būtų galima panaudoti bevielio ryšio ar kitą duomenų perdavimo technologiją naudojančią HTTP protokolą.

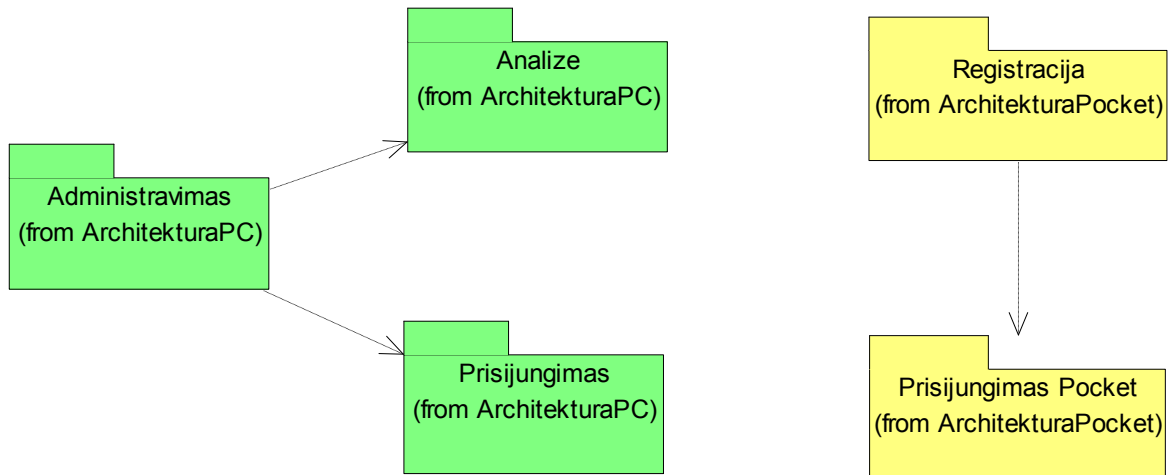
Parinkti tokią architektūrą, kurią būtų galima pritaikyti panašaus pobūdžio sistemoms (Mobilioms rungtynių rezultatų surinkimo ir analizės sistemoms pvz.: Rankinio varžybų.).

Architektūrinius apribojimus lemia pasirinktos sistemos kūrimo priemonės: *.NET* technologija paremta, objektinio programavimo kalba *C#*. Sistemos eksploatavimui reikalingas kompiuteris su *Microsoft® Windows® XP* ir delninis kompiuteris su *Microsoft® Windows® CE* (arba kita operacine sistema, kuri palaiko *Microsoft .NET Compact Framework*) operacinėmis sistemomis. Turi būti įdiegtas *Microsoft .NET Framework* ir *Microsoft .NET Compact Framework*, atitinkamai personaliniam ir delniniam kompiuteriams. Papildomai sistemos funkcionavimui turi būti įdiegta MSDE duomenų bazių valdymo sistema pagrindiniame serveryje ir *Microsoft Sql Server CE* delniniame kompiuteryje.

3.4 SISTEMOS STATINIS VAIZDAS

3.4.1 APŽVALGA

Kuriama sistema veiks skirtingose aplinkose: asmeniniame kompiuteryje bei delniniame kompiuteryje. Todėl, projektuojant sistemą, buvo išskirtos nepriklausomos paketų grupės. Nustatyti programinės įrangos paketai bei jų priklausomybės yra pateikiamos 17 paveiksle:



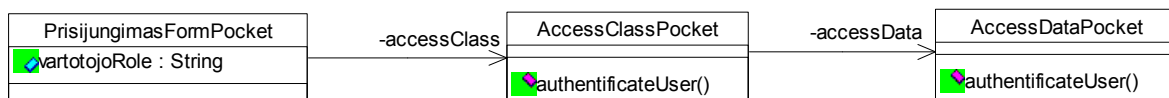
17 pav. Sistemos išskaidymas į paketus

Sistemos paketai *Administravimas*, *Analize* ir *Prisijungimas* atitinka sistemos dalį, kuri bus realizuojama asmeniniam kompiuteriui. Sistemos paketai *Registracija* ir *Prisijungimas Pocket* atitinka sistemos dalį, kuri bus realizuojama delniniame kompiuteryje.

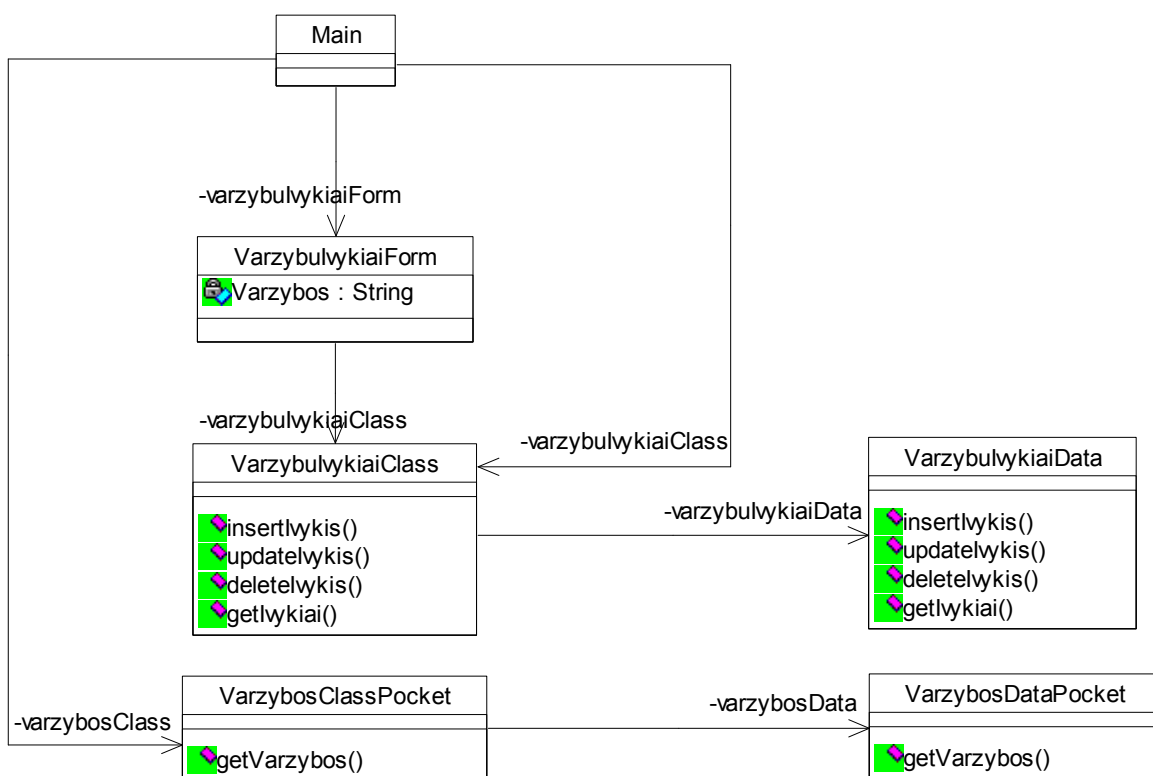
3.4.2 PAKETŲ DETALIZAVIMAS

Pateikiamas trumpas delninio kompiuterio sistemos paketų aprašymas:

- *Registracija* – sistemos dalis, kurios pagalba delniniame kompiuteryje yra registruojami krepšinio rungtynių rezultatai. Šis paketas naudoja paketą *Prisijungimas Pocket*
- *Prisijungimas Pocket* – sistemos dalis, kurios pagalba yra atliekamas prisijungimas delniniame kompiuteryje.
- Kiekvienam paketui yra sudaryta klasių diagrama. Paketų klasių diagramos pateikiamos 18 – 19 paveiksluose



18 pav. Paketo Prisijungimas Pocket klasių diagrama



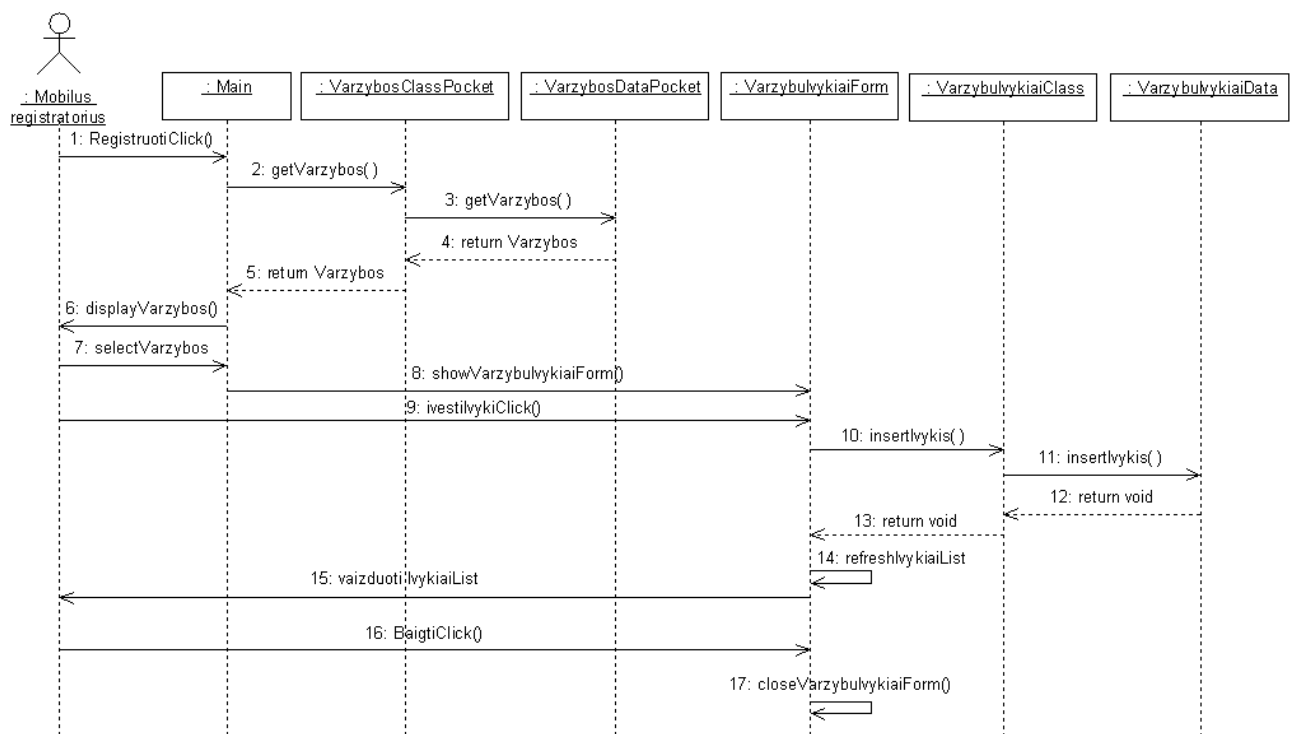
19 pav. Paketo Registracija klasių diagrama

3.4.3 SISTEMOS DINAMINIS VAIZDAS

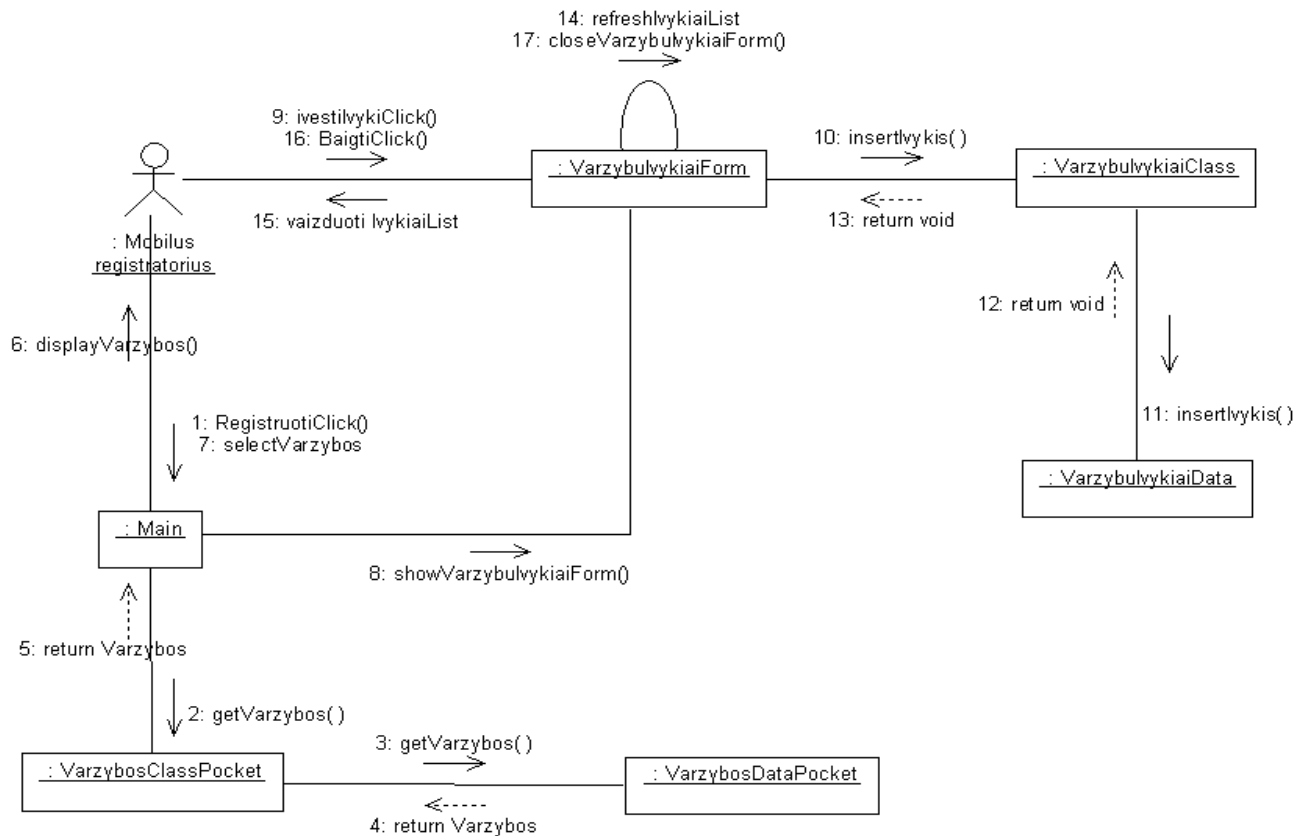
Sistemos dinaminio vaizdo pateikimui bus naudojamos veiklos, sekų, bendradarbiavimo bei būsenų kaitos diagramos.

Veiklos diagramos padeda atskleisti sistemos, posistemės arba komponento logiką. Šiame dokumente veiklos diagramos pateikiamos daugiau atsižvelgiant į panaudojimo atvejus, kurie yra paskirstyti po sistemos paketus.

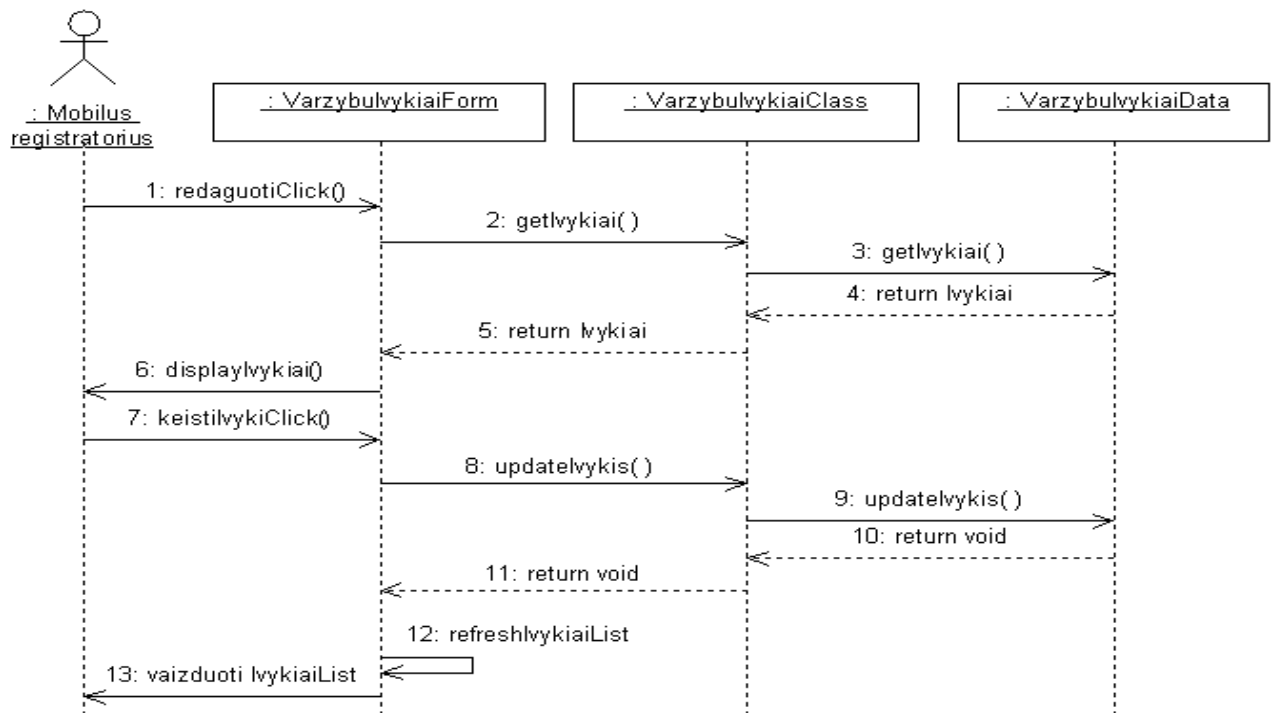
Registracijos posistemės sąveikos diagramos pateikiamos 20 – 24 paveiksluose.



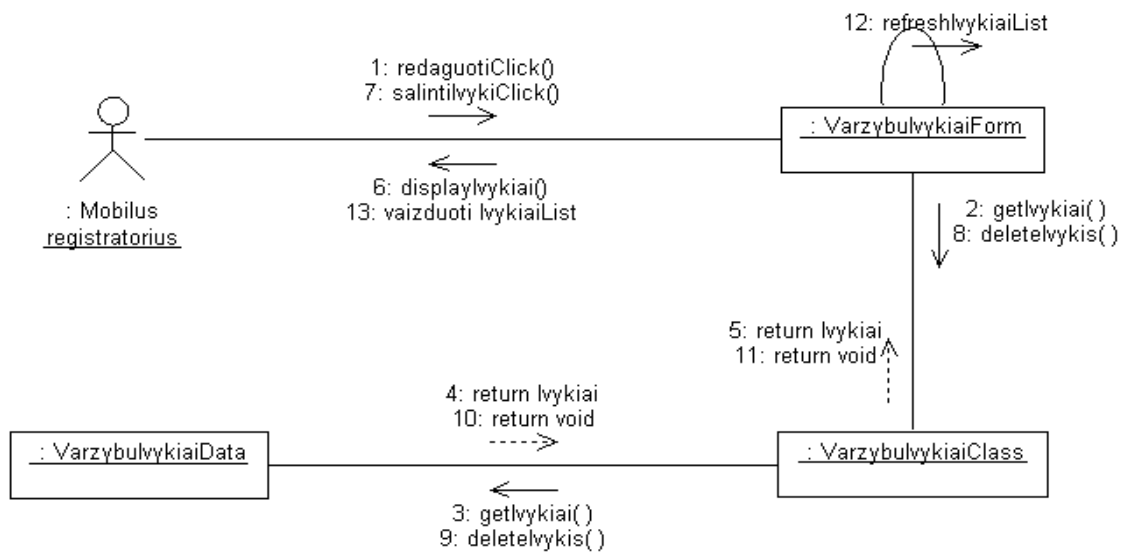
20 pav. Krepšinio varžybų įvykių registravimo sekų diagrama



21 pav. Krepšinio varžybų įvykių registravimo bendradarbiavimo diagrama

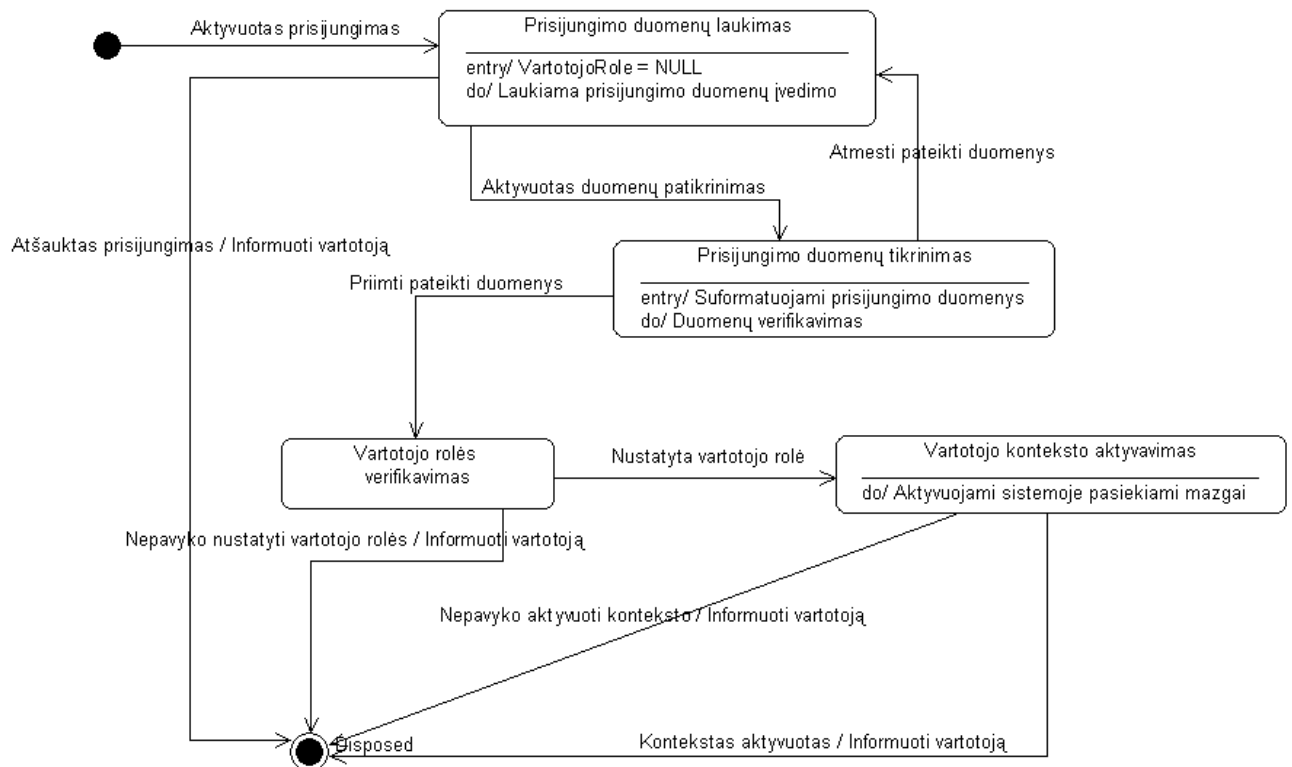


22 pav. Krepšinio varžybų įvykių šalinimo sekų diagrama



23 pav. Krepšinio varžybų įvykių šalinimo bendradarbiavimo diagrama

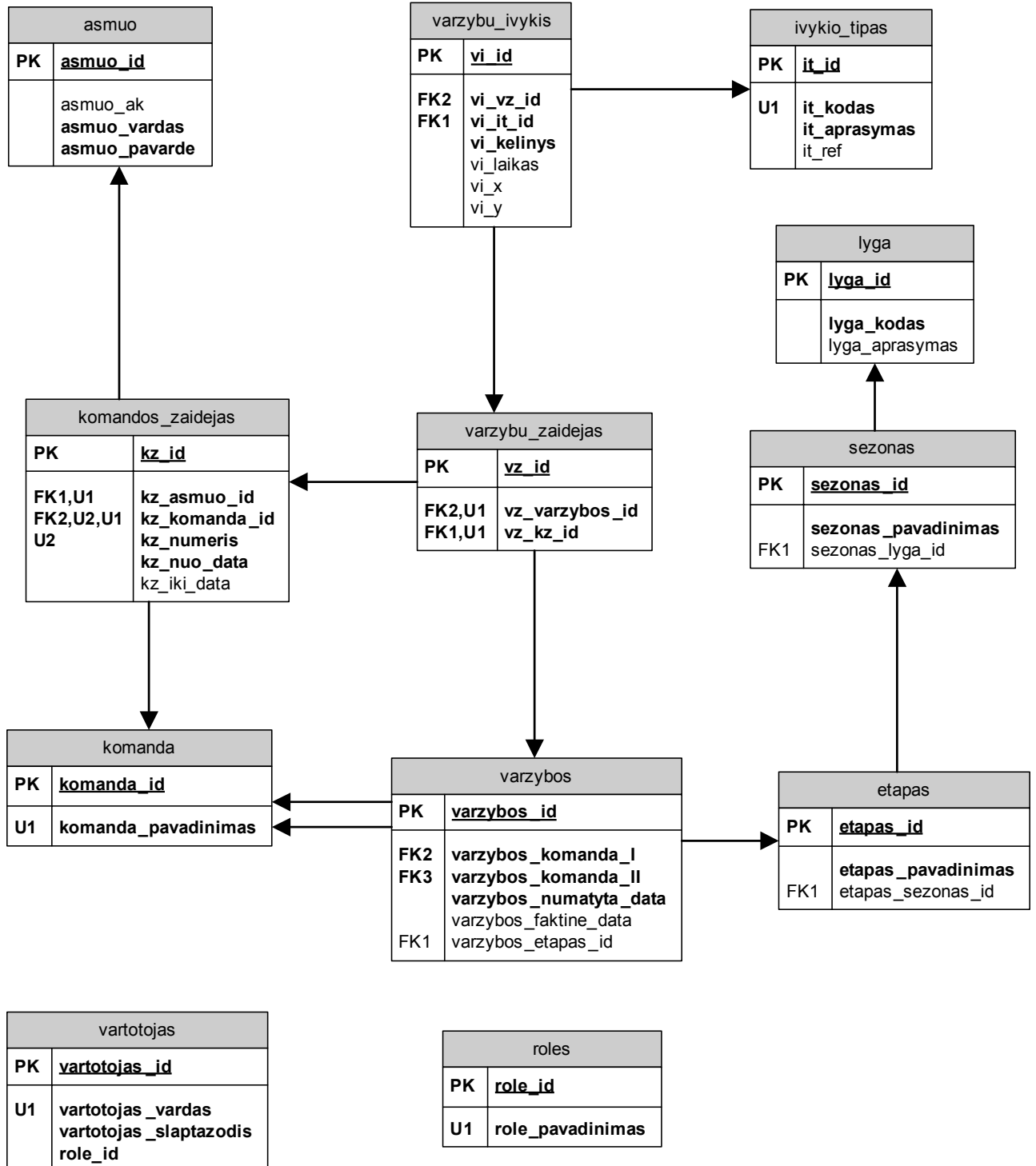
Sistemos dinaminio vaizdo pateikime taip pat naudojama būsenų kaitos diagrama. Būsenų kaitos diagrama parodo kaip keičiasi tam tikros klasės objekto būsenos. *Prisijungimas Pocket* klasės, esančios pakete *Prisijungimas* būsenų kaitos diagrama pateikiama 24 paveiksle.



24 pav. *Prisijungimas Pocket* klasės būsenų kaitos diagrama

3.5 DUOMENŲ VAIZDAS

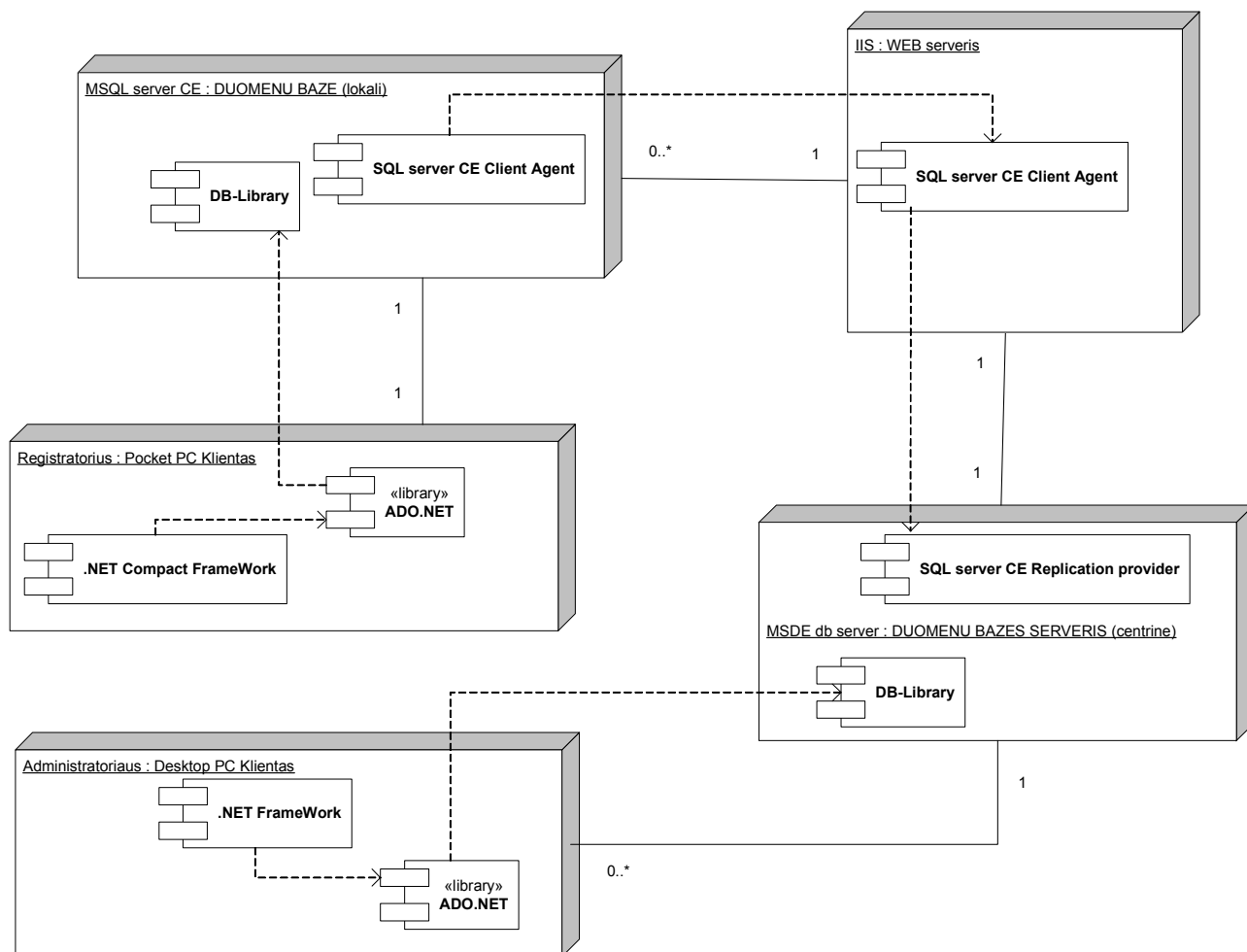
Žemiau pateiktas sistemos duomenų bazės modelis:



25 pav. Duomenų bazės modelis

3.6 IŠDĖSTYMO VAIZDAS

Žemiau pateiktas sistemos komponentų ir išdėstymo modelis:



26 pav. Išdėstymo diagrama su komponentais

4. EKSPERIMENTINIS SISTEMOS TYRIMAS

Tiriamoji dalis skirta realizuotos mobilios krepšinio varžybų įvykių registravimo ir analizės programinės įrangos tobulinimo galimybėms ir kokybės analizei.

4.1 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 eksperimentai:

- ✓ ištirti ar garsinių pranešimų panaudojimas padeda greičiau įvesti duomenis naudojant vienodo dydžio mygtukus;
- ✓ sužinoti ar mažesnio dydžio mygtukų naudojimas su garsiais pranešimais yra efektyvesnis negu didesnių mygtukų naudojimas be garsinio pranešimų;
- ✓ ištirti ar garsinių pranešimų panaudojimas sumažina įvykių įvedimo klaidų kiekį.

4.2 EKSPERIMENTO OBJEKTAS

Eksperimento metu bus tiriama delninio kompiuterio mobili krepšinio įvykių registravimo sistema.

4.3 EKSPERIMENTINIO TYRIMO EIGA

Vykdamas eksperimentą buvo sukurtos trys vienodos vartotojo sąsajos su skirtingais mygtuko dydžiais pavaizduota 7 lentelėje ir komponentas mygtukų paspaudimo trukmei skaičiuoti. Eksperimento metu buvo naudojamas *Compaq Pocket PC iPAQ H3950* delninis kompiuteris, kurio savybės aprašytos 1 lentelėje.

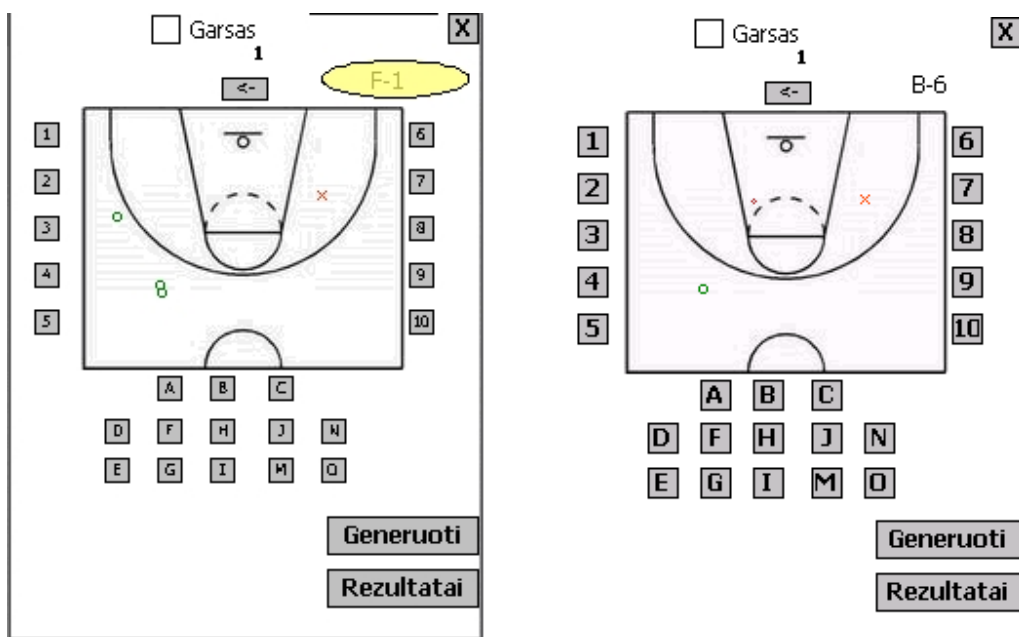
7 lentelė. Eksperimente naudojami mygtukų dydžiai

Mygtuko dydis, taškeliais	Be garso	Su garsu
12x12	x	x
16x16	x	x
20x20	x	x

Realizuota programų sistema buvo pateikta naudojimui. Eksperimento metu dalyvavo

šešiolika, skirtingo amžiaus dalyvių. Pagal amžių dalyviai pasiskirstė taip: iki 18m. – 0, 18-23m. – 4, 24-30m. – 5, 31-40m. – 4, 41-50m. – 2, 51-60m. – 1, 61 ir daugiau – 0. Jiems buvo pateikta mobili registravimo sistema, kurią galėjo išbandyti 5 minutes prieš kiekvieną eksperimentą. Pagrindinis atrankos kriterijus buvo mokėjimasis naudotis kompiuteriu.

Dalyviai turėjo registruoti atsitiktinai generuojamus varžybų įvykius. Generuojama buvo 50 atsitiktinai parinktų įvykių. Šiuos įvykius reikia įvesti naudojant skirtingo dydžio mygtukus ir panaudojant garsinius pranešimus arba be jų. Teisingam įvykių registravimui pirmiausia reikia paspausti reikiama raide pažymėtą mygtuką, paskui skaičiumi ir pažymėti vietą aikštelėje. Įvykių kombinacija, kurią reikia įvesti pažymėta geltonai, parodyta 27 paveiksluke.



27 pav. Vartotojo sąsajos su 12x12 ir 16x16 taškelių dydžių mygtukais

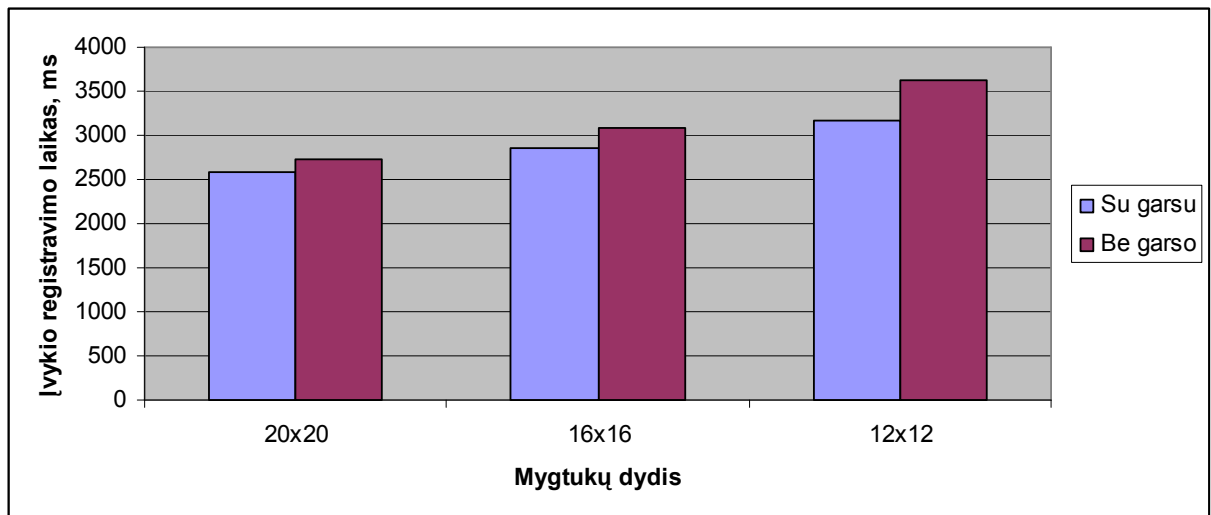
Tiriant įvykių registravimo laikus į programos kodą buvo įterpti procesoriaus laiko nuskaitymo operatoriai ir sukurtas šių duomenų apdorojimo komponentas. Laikas matuojamas suformavus ir pateikus reikiamą įvykį ir po įvykio registravimo. Šių dviejų laikų skirtumas atspindi pilną įvykio registravimo laiką.

Eksperimento rezultatai, kai naudojami skirtingo dydžio mygtukai ir garsiniai pranešimai:

8 lentelė. Įvykio registravimo laikų palyginimas

Garsinių pranešimų naudojimas	Mygtuko dydis		
	20x20	16x16	12x12
	Vidutinis įvykio įvedimo laikas, ms		
Naudoja	2576,4	2859,4	3176,6
Nenaudoja	2738,4	3082	3626,8

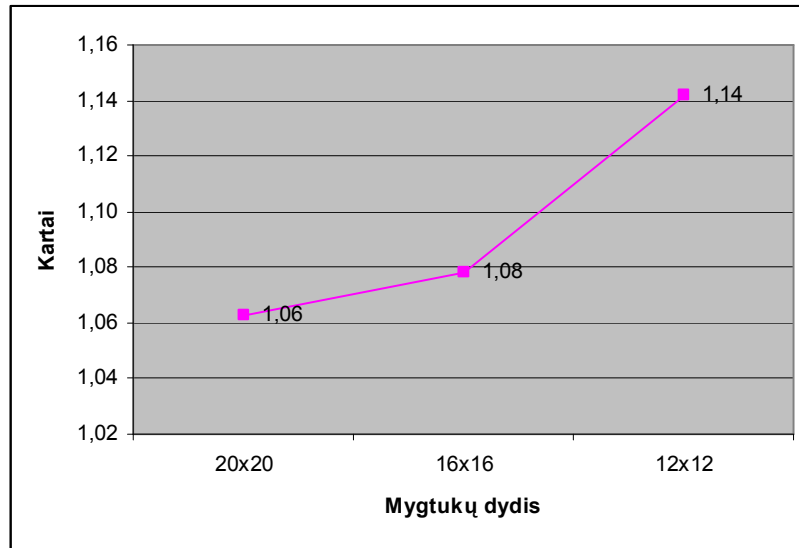
Naudojant skirtingo dydžio mygtukus ir garsinius pranešimus įvykių registravimo laikų diagrama:



28 pav. Vidutinių įvykio registravimo laikų palyginimas

Diagramoje matyti, jog įvykių registravimas, naudojant to paties dydžio mygtukus su garsiniais pranešimais leidžia greičiau įvedinėti duomenis nei nenaudojant garsinių pranešimų.

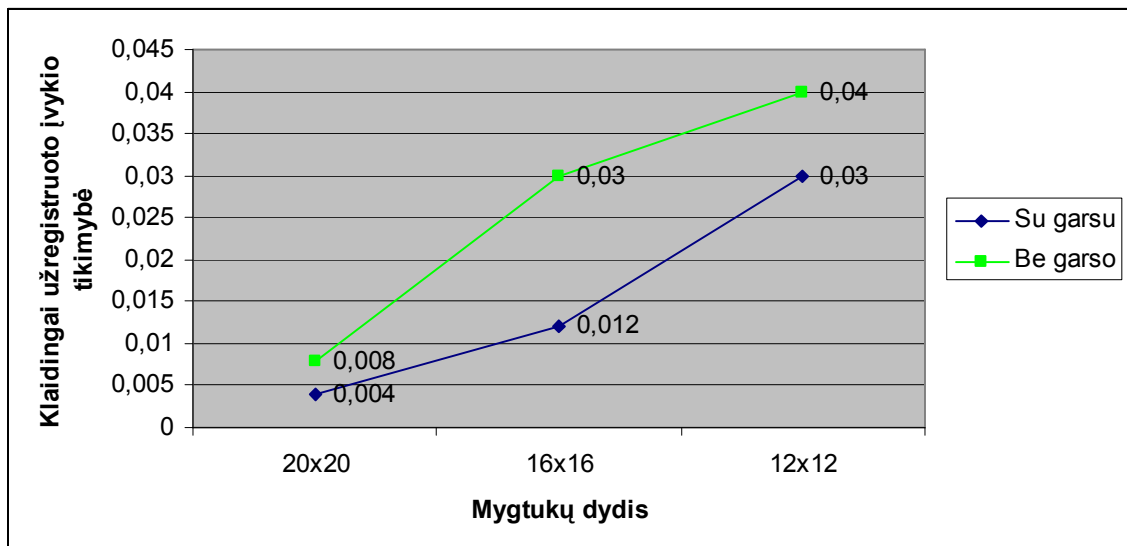
Duomenų įvedimas 1,1 karto yra efektyvesnis kai naudojami garsiniai pranešimai. Garsinių pranešimų ir be jų santykinų duomenų įvedimo greičių palyginimo diagrama parodo, kiek kartų efektyviau yra naudoti garsinius pranešimus:



29 pav. Santykinių greičių palyginimo diagrama naudojant garsinius pranešimus ir be jų

Klaidingų duomenų taisymui reikia laiko, todėl svarbu įvesti teisingus duomenis, kurių nereikėtų keisti. Eksperimento metu buvo tiriama ar garsiniai pranešimai sumažina klaidų kiekį įvedant duomenis nei nenaudojant garsinių pranešimų.

Naudojant garsinius pranešimus klaidingų duomenų įvedimo tikimybė to paties dydžio mygtukams yra mažesnė. Taigi garsinių pranešimų panaudojimas padeda sumažinti klaidingų duomenų įvedimo tikimybę, eksperimento rezultatai pavaizduoti diagramoje.



30 pav. Klaidingų duomenų įvedimo tikimybė

4.4 KOKYBĖS KRITERIJŲ VERTINIMAS PAGAL ISO/IEC 9126 KOKYBĖS MODELĮ

Bendram kokybės įvertinimui pasirinktas ISO/IEC 9126 kokybės modelis.

Pagal šį modelį yra 6 kokybės vertinimo charakteristikos [11]:

1. Funkcionalumas (angl. *Functionality*)
2. Patikimumas (angl. *Reliability*)
3. Panaudojamumas (angl. *Usability*)
4. Efektyvumas (angl. *Efficiency*)
5. Pernešamumas (angl. *Portability*)
6. Priderinamumas (angl. *Maintainability*)

4.4.1 FUNKCIONALUMAS

- Tinkamumas (angl. *Suitability*) – produkto teikiamos funkcijos (operacijos) pilnai pateikia funkcionalumą, kuris yra reikalaujamas reikalavimų specifikacijos dokumente.
- Tikslumas (angl. *Accuratness*) – produkto pateikiami rezultatai atitinka rezultatus, kurie yra reikalaujami pagal reikalavimų specifikacijos dokumentą.
- Sąveika (angl. *Interoperability*) – Produkto sistemų (bei posistemų) bendradarbiavimas nėra pilnai ištestuotas.
- Sauga (angl. *Security*) – produkto saugumas yra užtikrinamas vartotojo įgaliojimų suteikimo pagalba. Realizuotas apsaugotas prisijungimas prie delninio kompiuterio kliento. Duomenų bazės prisijungimas taip pat apsaugomas slaptažodžiu.

4.4.2 PATIKIMUMAS

- Branda (angl. *Maturity*) – tikslesniam brandos įvertinimui reikalingas ilgesnis darbo su programine įranga stebėjimas. Per neilgą darbo su programine įranga laikotarpį didesnių programos nesėkmių (angl. *failure*) nepastebėta.
- Klaidų tolerancija (angl. *Fault tolerace*) – sistema tikrina įvedamų įvykių teisingumą, jei įvykis netinkamas apie tai praneša vartotojui.
- Atstatomumas (angl. *Recoverability*) – atstatymo tyrimas nebuvo atliekamas, kadangi šios galimybės sistemoje nenumatyta.

4.4.3 PANAUDOJAMUMAS

- Suprantamumas (angl. *Understandability*) – vartotojo sąsajai paaiškinti yra parašyta vartotojo dokumentacija.
- Apsimokymas (angl. *Learnability*) – atlikus testuotojų apklausą nustatyta, kad vartotojai nesunkiai išmoko naudotis programine įranga. Paruošta vartotojo dokumentacija buvo pakankamai išsami.
- Valdomumas (angl. *Operability*) – daugumą sistemos funkcijų galima atlikti pelės arba aktyvaus ekrano pagalba. Tai vienas patogiausių valdymo būdų.

4.4.4 EFEKTYVUMAS

- Laikinė charakteristika (angl. *Time behavior*) – rezultatai pateikti 4.3 skyriuje.
- Resursai (angl. *Resource behavior*) – sistema apkrauna kompiuterio resursus minimaliai:
 - Procesoriaus apkrovimas: iki 40 %
 - Operatyvinė atmintis: apie 2 MB
 - Klientinės programos dydis: 0.6 MB

4.4.5 PERNEŠAMUMAS

- Adaptacija (angl. *Adaptability*) – programų sistema kurta konkrečiai aplinkai, todėl norint pritaikyti ją skirtingai aplinkai reikalingos papildomos priemonės ir veiksmai.
- Įdiegiamumas (angl. *Installability*) – kartu su programų sistema pateikiamos visos reikalingos priemonės pilnam įdiegimui. Įdiegimas detaliai aprašytas vartotojo dokumentacijoje.

4.4.6 PRIDERINAMUMAS

- Analizuojamumas (angl. *Analyzability*) – dėl didelio ryšių tarp sistemos komponentų kiekio, gali būti sudėtinga nustatyti gedimo vietą arba modifikuojamą programos bloką.
- Modifikuojamumas (angl. *Changeability*) – sistema sukurta objektine programavimo kalba, todėl keisti tam tikras kodo dalis nėra sudėtinga.
- Stabilumas (angl. *Stability*) – kodo pakeitimas gali iššaukti sistemos nestabilumą.
- Testuojamumas (angl. *Testability*) – didelis ryšių kiekis tarp programinės įrangos komponentų, todėl reikia daug pastangų atliekant integravimo testavimą.

5. IŠVADOS

1. Buvo išanalizuotos delninių kompiuterių programinės įrangos vartotojo sąsajos problemos, pateikti galimi problemų sprendimo būdai.
2. Darbe buvo ištirtos ir palygintos delninių ir asmeninių kompiuterių jau sukurtos krepšinio varžybų įvykių registravimo ir analizės sistemos.
3. Išnagrinėtos ir palygintos *Microsoft Windows 32-Bit API* ir *.Net Compact Framework* programavimo technologijos.
4. Pasirinktomis programavimo technologijomis (*Microsoft .NET*) buvo suprojektuota ir sukurta mobili krepšinio varžybų registravimo ir analizės sistema.
5. Atliktas eksperimentas patvirtino, kad garsinių pranešimų panaudojimas mygtuko paspaudimui nustatyti, padidina registruojamų įvykių įvedimo greitį ir sumažina klaidos tikimybę.
6. Kitos darbe realizuotos sistemos charakteristikos (funkcionalumas, patikimumas, efektyvumas, pernešamumas ir priderinamumas) ištyrinėtos panaudojant ISO/IEC 9126 kokybės modelį.

6. LITERATŪRA

- [1] **Whitten, J.; Bentley, L.; ir Dittman, K.** *Systems Analysis and Design Methods 6th ed.*, McGraw-Hill, 2004, p. 470 -684. ISBN 0131454552.
- [2] **Laberge, R.; ir Vujosevic, I.** *Building PDA Databases for Wireless and MobileDevelopment* Canada: Wiley Publishing, Inc., 2003, ISBN 0-471-21645-3.
- [3] **Fox, G.; Kim, K.; ir Lee, S.** *Integration of Hand-Held Devices into Collaborative Environments*, USA: University of Indiana, 2002.
- [4] **Wilding-McBride, D.** *Java™ Development on PDAs: Building Applications for PocketPC and Palm Devices*, Addison Wesley, 2003, ISBN 0-201-71954-1.
- [5] **Fox, G.; Ko, S.; ir Lee, S.** *Status of Hand-Held Interface to Garnet Collaborative Environment*, USA: University of Indiana, 2002.
- [6] **Yacoub, S.; ir Ammar, H.** *Pattern-Oriented Analysis and Design: Composing Patterns to Design Software Systems*, Addison Wesley, 2003, ISBN 0201776405.
- [7] Delninių kompiuterių palyginimo [interaktyvus]. 2005 [žiūrėta 2006-02-27]. Prieiga per internetą:
<http://www.pcworld.com/reviews/chart_comparison.asp?aid=116027&prodid=19269&prodid=21213&prodid=21462&prodid=21249#>.
- [8] Delninių kompiuterių programinės įrangos kūrėjų tinklapis [interaktyvus]. 2005 [žiūrėta 2006-02-17]. Prieiga per internetą: <http://msdn.microsoft.com/library/en-us/dnppcgen/html/nw_pocket_analyze_decision_support_wm2k3_ppc.asp>.
- [9] **Ren, X.; ir Moriya, S.** *The best among six strategies for selecting a minute target and the determination of the minute maximum size of the targets on a pen-based computer.* Chapman & Hall, 1997, pp. 85-92
- [10] **Wigley, A.; ir Roxburgh, P.** *Building .NET Applications for Mobile Devices Editor* , Microsoft Press, 2002, ISBN 073561914X.
- [11] ISO 9126: The Standard of Reference. Iš HCI Bibliography : iš CSE – Centre for Software Engineering [interaktyvus]. 2004, kovas [žiūrėta 2005-11-20], prieiga per internetą: <<http://www.cse.dcu.ie/essscope/sm2/9126ref.html>>
- [12] **Fox, D.; ir Fox, J.** *Deployment Patterns for Microsoft .NET Compact Framework.* Iš MSDN [interaktyvus]. 2004, lapkritis. [žiūrėta 2005-12-15]. Prieiga per internetą: <<http://msdn.microsoft.com/library/en-us/dnnetcomp/html/DeploymentPatterns.asp>>.
- [13] *PocketStats* tinklalapis [interaktyvus]. 2005 [žiūrėta 2006-01-25]. Prieiga per internetą: <http://www.tekriver.com/products/psbskt/psbskt_ss.html#PocketStats>.
- [14] **Karlson, K.; ir Bederson, B.** *AppLens and LaunchTile: Two Designs for One-Handed Thumb Use on Small Devices.* Iš Small Devices 1 [interaktyvus]. 2005, balandis. [žiūrėta 2005-11-25]. Prieiga per internetą:<<http://www.cs.umd.edu/hcil/piccolo/learn/papers/ch-applens-launchtile.pdf>>.
- [15] *TurboStats* tinklalapis [interaktyvus]. 2004 [žiūrėta 2006-01-22]. Prieiga per internetą: <<http://www.turbostats.com/basketsk.html>>.
- [16] **YAO, P.** *Choosing a Windows Embedded API: Win32 vs. the .NET Compact Framework* [interaktyvus]. The Microsoft Developer Network, September 2002 [žiūrėta 2004. lapkričio 4 d.]. Prieiga per internetą: <http://msdn.microsoft.com/library/en-us/dncenet/html/choose_api.asp?frame=true>.

- [17] **Gong, J.; ir Tarasewich, P.** *Guidelines for Handheld Mobile Device Interface Design* [interaktyvus]. 2003, lapkritis [žiūrėta 2005-09-29]. Prieiga per internetą: <<http://www.ccs.neu.edu/home/tarase/GuidelinesGongTarase.pdf>>.
- [18] **Pierce, K.; ir Sinclair, J.** *Sensing Techniques for Mobile Interface*. [interaktyvus]. 2002 [žiūrėta 2004. lapkričio 4 d.]. Prieiga per internetą: <<http://www.alice.org/stage3/pubs/uistsensing.pdf>>
- [19] **Kiyota, K.; ir Takizava H.** *A Circular Fashion Menu System Based on Human MotorControl Knowledge for the Pen-based Computer*. [interaktyvus]. 2003 [žiūrėta 2005. lapkričio 12 d.]. Prieiga per internetą: <<http://www.info.toba-cmt.ac.jp/~ezaki/doc/hci2003-Kiyota.pdf>>

7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

API – taikomųjų programų kūrimo sąsaja (angl. *Application Programming Interface*).

CPU – centrinis procesorius (angl. *central processing unit*).

DB – duomenų bazė.

DBVS – duomenų bazių valdymo sistema.

DLL – dinaminė biblioteka (angl. *dynamic – link library*).

HTTP – hiperteksto perdavimo protokolas (angl. *hypertext transfer protocol*).

HTTPS – saugus hiperteksto perdavimo protokolas (angl. *Hyper Text Transfer Protocol Secure sockets*).

IDE – integruota kūrimo aplinka (angl. *Integrated Development Environment*).

IIS – interneto duomenų serveris (angl. *Internet Information Server*).

PA – panaudojimo atvejai.

PAM – panaudojimo atvejų modelis.

PDA – asmeninis skaitmeninis asistentas (angl. *Personal Digital Assistant*).

PĮ – programinė įranga.

PS – programų sistema.

PC – asmeninis kompiuteris.

RPC – nutolusių procedūrų kvietimas (angl. *Remote Procedure Calls*).

RUP – racionalus unifikuotas procesas (angl. *Rational Unified Process*).

SOAP – paprastas objektų priėjimo protokolas (angl. *Simple Object Access Protocol*).

SQL – struktūrizuota užklausų kalba (angl. *Structured Query Language*).

SSL – saugių sujungimo mazgų lygmuo (angl. *Secure Sockets Layer*).

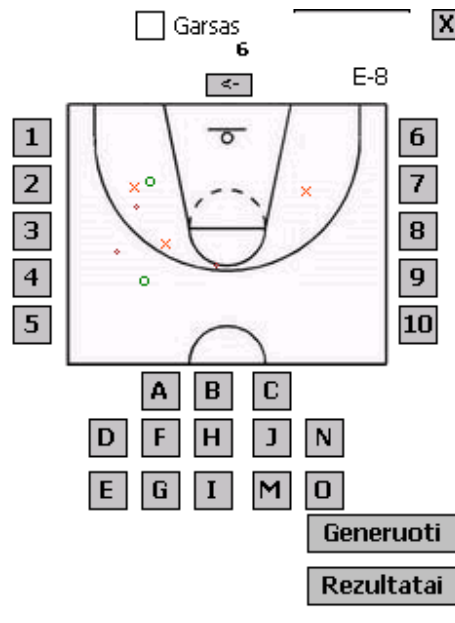
TCP – perdavimo valdymo protokolas (angl. *Transmission Control Protocol*).

UML – unifikuota modeliavimo kalba (angl. *Unified Modeling Language*).

XML – praplečiama žymių kalba (angl. *Extensible Markup Language*).

XML žiniatiklio paslaugos – aplikacijos komponentas, prieinamas per standartinius interneto protokolus (angl. *XML Web services*).

8. PRIEDAI



31 pav. Vartojimo sąsaja su 20x20 mygtukų dydžiais



32 pav. Vartojimo sąsaja prieš eksperimentą

9 lentelė. Eksperimento rezultatai

Mugtyko dydis, taškeliais	20x20			16x16			12x12			
	Teisingi	Klaidos	Vidutinis ivykio registravimo laikas, ms	Teisingi	Klaidos	Vidutinis ivykio registravimo laikas, ms	Teisingi	Klaidos	Vidutinis ivykio registravimo laikas, ms	
Edita Paulavičiūtė	Be garso	49	1	2733	47	3	3111	47	3	3777
	Su garsu	49	1	2556	48	2	2864	48	2	3242
Nerijus Černiauskas	Be garso	49	1	2721	49	1	3095	48	2	3656
	Su garsu	50	0	2501	50	0	2902	50	0	3203
Nerijus Skruodys	Be garso	49	1	2822	49	1	3203	48	2	3744
	Su garsu	50	0	2705	50	0	3101	49	1	3244
Rolandas Anisovas	Be garso	50	0	2660	50	0	2905	50	0	3578
	Su garsu	50	0	2580	50	0	2786	50	0	3108
Egidijus Papaurėlis	Be garso	50	0	2756	47	3	3085	47	3	3599
	Su garsu	50	0	2540	49	1	2733	49	1	3159
Saulius Pakriajauskas	Be garso	49	1	2834	47	3	3172	46	4	3711
	Su garsu	50	0	2677	49	1	2996	49	1	3255
Nora Sakalauskaite	Be garso	50	0	2739	47	3	3131	47	3	3429
	Su garsu	50	0	2520	49	1	2820	48	2	3045
Tomas Janusauskas	Be garso	49	1	2688	47	3	3014	48	2	3479
	Su garsu	50	0	2540	49	1	2852	49	1	3086
Ramūnas Keliuotis	Be garso	50	0	2733	50	0	3005	49	1	3459
	Su garsu	50	0	2535	50	0	2700	49	1	3116
Rūta Sakalauskaite	Be garso	49	1	2793	50	0	3085	47	3	3479
	Su garsu	49	1	2577	50	0	2838	49	1	3186
Robertas Barisevičius	Be garso	50	0	2750	49	1	3053	49	1	3566
	Su garsu	50	0	2587	49	1	2833	49	1	3211
Agnė Paulavičiūtė	Be garso	50	0	2721	50	0	3092	50	0	3730
	Su garsu	50	0	2569	50	0	2877	49	1	3280
Aurimas Ožekauskas	Be garso	50	0	2700	47	3	3085	48	2	3664
	Su garsu	50	0	2540	49	1	2733	49	1	3098
Andrius Mačys	Be garso	50	0	2737	50	0	3104	49	1	3723
	Su garsu	50	0	2572	50	0	2874	49	1	3196
Mantas Kukenis	Be garso	49	1	2678	49	1	3039	48	2	3697

	Su garsu	49	1	1	2575	49	1	1	2897	50	1	3173
Ramūnas Čerņiauskas	Be garso	50	0	0	2750	47	3	3	3133	47	3	3738
	Su garsu	49	1	1	2649	49	1	1	2945	49	2	3223
Vidurķiai:	Be garso	49,6	0,4	0,4	2738,4	48,4	1,6	1,6	3082,0	48,0	2,0	3626,8
	Su garsu	49,8	0,2	0,2	2576,4	49,4	0,6	0,6	2859,4	49,2	1,2	3176,6