

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

MANTAS VAITKŪNAS

PHP KARKASŲ ARCHITEKTŪRINIS TYRIMAS

Magistro darbas

Darbo vadovas
doc. dr. T. Blažauskas

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ STUDIJŲ PROGRAMA

MANTAS VAITKŪNAS

PHP KARKASŲ ARCHITEKTŪRINIS TYRIMAS

Magistro darbas

Darbo vadovas:
doc. dr. T. Blažauskas
2013-05-23

Recenzentas:
lekt. dr. Arūnas Tomkevičius
2013-05-23

Atliko:
IFM-1/2 gr studentas
Mantas Vaitkūnas
2013-05-23

KAUNAS, 2013

AUTENTIŠKUMO PATVIRTINIMAS

Patvirtinu, kad įteikiamas baigiamasis darbas „PHP karkasų architektūrinis tyrimas“ yra:

1. Autoriaus atliktas savarankiškai, jame nėra pateikta kitų autorių medžiagos kaip savos, nenurodant tikrojo šaltinio.
2. Nebuvo to paties autoriaus pristatytas ir gintas kitoje mokymo įstaigoje Lietuvoje ar užsienyje.
3. Nepateikia nuorodų į kitus darbus, jeigu jų medžiaga nėra naudota darbe.
4. Pateikia visą naudotos literatūros sąrašą.

Data, studento vardas, pavardė, parašas

SANTRAUKA

Šiuo metu yra sukurta daugybė PHP karkasų, tačiau yra mažas informacijos kiekis subjektyviai nusakančios privalumus ir trūkumus. Pagrindinis darbo tikslas yra ištirti Yii, Zend 2 ir Symfony2 karkasus, bei juos palyginti.

Šiame dokumente aprašytas darbas susideda iš trijų pagrindinių dalių. Pirmoji dokumento dalis – karkasų apžvalga. Šioje dalyje apžvelgiami Yii, Zend 2 ir Symfony2 karkasai. Antroje (projektinėje) dokumento dalyje aprašoma sistema, kuri yra kuriama atskirai su kiekvienu iš karkasų. Šiame skyriuje pateikiama visa sistemos architektūra, reikalavimai sistemai, bendri apribojimai, statinis vaizdas. Trečioje (tyrimo ir eksperimentinėje) dokumento dalyje, detalizuojami atlikti tyrimai ir eksperimentai.

Tyrimai vykdomi kuriant QR kodų generavimo ir vartotojų lankomumo statistikų sekimo sistemą. Šių tyrimų metu tiriamas darbo imlumas, našumas, funkcionalumas ir atlikta pasirinktų karkasų kodo analizė.

PHP FRAMEWORKS ARCHITECTURAL RESEARCH

SUMMARY

Currently there are created plenty of different PHP frameworks, but the amount of information subjectively outlining the pros and cons is low. Main purpose of this research is to investigate and compare three main frameworks: Yii, Zend 2 and Symfony.

This document describes work consisting of three main parts. The first part of the document is a framework review. This section provides an overview of Yii, Zend 2 and Symfony2 frameworks. Second (design) part is a description of the system which was developed using each of the frameworks. This section contains the entire system architecture, system requirements, general restrictions and static image. The third (research and experimental) part of the document, consists of detailed studies and experiments.

Studies were carried out in developing QR code generation and visitor monitoring system. During these studies the capacity, performance, functionality and code of three main frameworks were analyzed.

TURINYS

Terminų ir santrumpų žodynas.....	10
1 Įvadas.....	11
2 Karkasų apžvalga ir analizė.....	12
2.1 MVC šablonas.....	13
2.2 Yii karkaso apžvalga.....	14
2.2.1 Yii architektūrinis vaizdas.....	14
2.2.2 Yii karkaso įrankiai.....	14
2.2.3 Yii karkaso trūkumai.....	15
2.3 Zend 2 karkaso apžvalga.....	15
2.3.1 Zend 2 karkaso architektūrinis vaizdas.....	15
2.3.2 Zend 2 karkaso ypatybės.....	16
2.3.3 Zend 2 karkaso trūkumai.....	16
2.4 Symfony2 karkaso apžvalga.....	16
2.4.1 Symfony2 karkaso paketai.....	16
2.4.2 Symfony2 karkaso įrankiai.....	16
2.4.3 Symfony2 karkaso trūkumai.....	17
3 QR kodų generavimo ir statistikų pateikimo sistemos kūrimas, panaudojant skirtingus karkasus, projektinė dalis.....	18
3.1 QR kodų generavimo ir statistikų pateikimo sistemos tikslai.....	18
3.2 Bendri apribojimai.....	18
3.3 Sistemos panaudojimų atvejų vaizdas.....	19
3.4 Sistemos statinis vaizdas.....	27
3.5 Sistemos veiklos diagramos.....	28
3.6 Sistemos būsenų diagramos.....	30
3.7 Sistemos sekų diagramos.....	32
3.8 Sistemos duomenų vaizdas.....	35
4 PHP karkasų tyrimo ir eksperimentinė dalis.....	36
4.1 Tyrimuose naudojamos sistemos parametrai.....	36
4.2 PHP karkasų darbo imlumo tyrimas.....	36
4.2.1 Yii karkaso darbo imlumo tyrimas.....	36
4.2.2 Zend 2 karkaso darbo imlumo tyrimas.....	37
4.2.3 Symfony2 karkaso darbo imlumo tyrimas.....	37
4.3 PHP karkasų našumo tyrimas.....	38
4.3.1 Našumo tyrimo rezultatai.....	38
4.4 Karkasų funkcionalumo tyrimas.....	39

4.5 Karkasų kodo analizė.....	40
4.5.1 Kodo analizės tyrimo aprašymas.....	41
4.5.2 Yii karkaso kodo analizės tyrimo rezultatai.....	43
4.5.3 Zend 2 karkaso kodo analizės tyrimo rezultatai.....	43
4.5.4 Symfony2 karkaso kodo analizės tyrimo rezultatai.....	44
5 Išvados.....	45
6 Literatūra.....	46

LENTELIŲ SĄRAŠAS

Lentelė 1: PA „Registruotis”	20
Lentelė 2: PA „Prisijungti”	21
Lentelė 3: PA „Kurti aplikaciją”	22
Lentelė 4: PA „Palyginti technologijas”	23
Lentelė 5: PA „Peržiūrėti statistikas”	24
Lentelė 6: PA „Skenuoti QR kodą”	25
Lentelė 7: PA „Turinio peržiūra”	26
Lentelė 8: Karkasų funkcionalumo skirtumai ir panašumai.....	39
Lentelė 9: Kategorijos- dydis ir sudėtingumas, kodo metrikos.....	41
Lentelė 10: Kategorijos- jungiamumas, kodo metrikos.....	41
Lentelė 11: Kategorijos- paveldėjimas, kodo metrikos.....	41
Lentelė 12: Metrikų atskaitinės vertės.....	42

PAVEIKSLŲ SĄRAŠAS

Pav. 1: MVC šablono grafinis vaizdas.....	13
Pav. 2: Yii karkaso architektūrinis vaizdas.....	14
Pav. 3: Zend 2 karkaso architektūrinis vaizdas.....	15
Pav. 4: Panaudojimo atvejų atvaizdas.....	19
Pav. 5: Paketų diagrama.....	27
Pav. 6: QR kodų generavimas ir aplikacijų kūrimas.....	28
Pav. 7: Aplikacijų pateikimas viešojoje dalyje.....	29
Pav. 8: Statistikų atvaizdavimas.....	29
Pav. 9: QR kodų generavimas ir WEB aplikacijų kūrimas.....	30
Pav. 10: Aplikacijų pateikimas viešojoje dalyje.....	30
Pav. 11: Statistikų atvaizdavimas.....	31
Pav. 12: QR kodų generavimas ir WEB aplikacijų kūrimas.....	32
Pav. 13: WEB aplikacijų pateikimas viešojoje dalyje.....	33
Pav. 14: Statistikų atvaizdavimas.....	34
Pav. 15: Duomenų bazės lentelė.....	35
Pav. 16: Užklausų kiekis per 30 sekundžių.....	38
Pav. 17: Procentaliai pateiktas užklausų aptarnavimas per tam tikrą laiką.....	39
Pav. 18: Apžvalgos piramidės struktūra.....	40
Pav. 19: Skaičiuojamos reikšmės peržiūros piramidėje.....	42
Pav. 20: Yii karkaso tyrimo rezultatai pateikti apžvalgos piramide.....	43
Pav. 21: Zend 2 karkaso tyrimo rezultatai pateikti apžvalgos piramide.....	43
Pav. 22: Symfony2 karkaso tyrimo rezultatai pateikti apžvalgos piramide.....	44

TERMINŲ IR SANTRUMPŲ ŽODYNAS

LAMP - Linux, Apache, MySQL, PHP serveris.

RPS (angl. Requests per seconds) - Užklausų kiekis per tam tikrą laiką.

APC (angl. Alternative PHP Cache) - PHP kešavimo karkasas.

MVC (angl. Model-View-Controller) - programinės įrangos architektūrinis modelis.

HTML (angl. Hyper text Markup Language) - kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete.

HTTP (angl. Hyper Text Transfer Protocol) - užklausimo - atsakymo protokolas, jungiantis klientą ir serverį.

ORM (angl. Object-relational mapping) – programavimo technika paverčianti tam tikrus duomenis į programavimo kalbos objektus.

DB (angl. Data Base) – duomenų bazė.

PA – panaudojimo atvejai.

WEB (angl. World Wide Web) - hiperteksto sistema, kuri veikia per internetą.

WEB aplikacija – aplikacija pasiekama vartotojo per interneto ar intraneto tinklą.

DBVS – duomenų bazės valdymo sistema.

QR kodas (angl. Quick Response Code) - dviejų dimensijų juodų taškų tinklelis, leidžiantis užkoduoti informaciją.

PHP (angl. Hypertext Preprocessor) - tai plačiai paplitusi dinaminė interpretuojama programavimo kalba.

Yii (angl. Yes It Is) – vienas iš populiariausių PHP karkasų.

CRUD (angl. Create Read Update Delete) – pagrindinės naudojamos operacijos: kūrimo, nuskaitymo, atnaujinimo ir trynimo.

PA – panaudojimo atvejis.

1 ĮVADAS

Šis dokumentas yra programų sistemų inžinerijos baigiamasis darbas. Jis yra skirtas programuotojams besirenkantiems PHP karkasą naujam projektui.

Šiuo metu yra sukurta daugybė PHP karkasų, tačiau yra mažas informacijos kiekis subjektyviai nusakančios privalumus ir trūkumus, todėl programuotojams renkantis karkasą tenka pasitikėti kolegų rekomendacijomis. Norint žinoti kurį karkasą išsirinkti konkrečiam projektui - reikia juos ištirti.

Pagrindinis projekto tikslas yra ištirti pasirinktų karkasų privalumus ir trūkumus, bei juos palyginti. Tikslui pasiekti reikia spręsti šiuos uždavinius:

- Apžvelgti Yii, Zend 2 ir Symfony2 karkasus.
- Suprojektuoti ir sukurti sistemą naudojant kiekvieną iš pasirinktų karkasų, kad būtų galima ištirti karkasų galimybes ir juos palyginti.
- Ištirti pasirinktų karkasų našumą, darbo imlumą, funkcionalumą, bei atlikti karkasų kodo analizę.
- Pateikti vykdytų tyrimų išvadas.

Atsižvelgus į atliktus tyrimus galima nustatyti karkasų silpnąsias ir stipriąsias vietas, juos palyginti ir pasirinkti tinkamiausią karkasą konkrečiam internetiniam projektui.

Šio darbo antrame skyriuje pateikiama karkasų apžvalga, sudedamosios dalys, jų įrankiai ir ypatybės. Trečiame skyriuje pateikiamas QR kodų generavimo ir statistikų pateikimo sistemos panaudojimo atvejų vaizdas, statinis vaizdas, veiklos diagramos, būsenų diagramos, sekų diagramos, bei duomenų vaizdas. Ketvirtame skyriuje pateiktas tyrimas atliktas kuriant QR kodų generavimo ir statistikų pateikimo sistemą. Penktame skyriuje pateiktos bendros darbo išvados.

2 KARKASŲ APŽVALGA IR ANALIZĖ

Karkasai suteikia pagrindą, kur programuotojų rašomas kodas išdėstomas tvarkingai. Dokumentacija, dažnai pasitaikančios problemos, sprendimai, patarimai ir konsultacijos pateikiamos internete atviro kodo stiliumi. Prieš karkasų naudojimą, didžioji dalis internetinių svetainių buvo kuriamos personalizuotai, kiekvienas programuotojas naudojo savo asmeninį technikos ir metodų arsenalą, tvarkant kodo išdėstymą ir kuriant sistemos architektūrą. Tai kėlė daug problemų: dokumentacijos trūkumą, sudėtingus karkasus ir bendrą klausimą, kas bus kai pasikeis programuotojas?

Karkasai buvo pradėti naudoti 2004 metais, kai buvo pristatyti kaip visapusiškas sprendimas kuriant stabilias internetines svetaines. Populiariausi PHP karkasai daugiausiai buvo skirti projektavimo metodologijai ir buvo naudojamas modelio-vaizdo-kontrolierius (MVC). Ko gero nėra sutapimas, kad 2004 metais buvo paleistas ir gerai žinomas Ruby on Rails MVC karkasas, kuris įkvėpė daug kitų PHP karkasų[1].

Šiuo metu PHP karkasų yra daug ir įvairių ir pasirinkti tinkamą yra sunku, tačiau dažniausiai naudojami karkasai yra užsitarnavę vartotojų pasitikėjimą ir internetinėje erdvėje apstu straipsnių ir techninės informacijos apie daugelį dažniausiai naudojamų karkasų.

Paprastai renkantis karkasą yra atsižvelgiama į šiuos kriterijus:

- Populiarumą
- Dokumentacijos aiškumą
- Greitį
- Vartotojų atsiliepimus internetinėje erdvėje
- Karkaso paprastumą
- Lankstumą

Šie kriterijai yra itin svarbūs kuriant sistemą ir reikalavimai jiems dažniausiai priklauso nuo sistemos tipo: ar tai būtų sudėtinga korporacinė sistema, ar paprasta informacinė svetainė.

Šiuo metu vieni iš dažniausiai naudojamų karkasų yra šie:

1. Yii- šis karkasas tapo pagrindiniu pasirinkimu kuriant saugią ir greitą aplikaciją, su mažesniu kodo eilučių kiekiu.
2. Zend 2 - pilnas pažangių PHP funkcijų suteikia geriausią aplinką PHP aplikacijoms kurti.
3. Symfony2 - karkasas, kurį sukūrė Prancūzų įmonė *SensioLabs*, yra išskirtinis. Šis karkasas skatina naudoti ORM, išsamiai apdoroja duomenų bazes ir naudoja savo paties šablonų variklį, taip pat sukurtą *SensioLabs*.

Kiekvienas iš šių karkasų paremtas MVC šablonu ir turi daug privalumų bei trūkumų, todėl norint išsirinkti tinkamiausią reikia iširti juos atsižvelgiant į kriterijus minėtus 1.1 skyriuje.

2.1 MVC šablonas

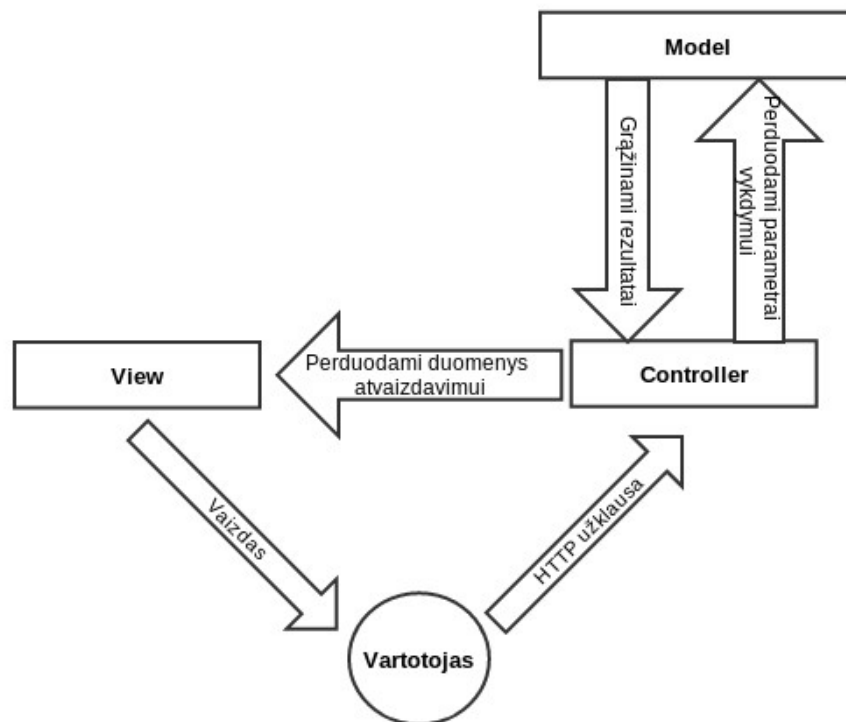
MVC - tai Model-View-Controller šablonas arba principas, vartotojui pateikiamą informaciją atskiriantis nuo sąveikos su ja. Šis šablonas padalina programą į trijų rūšių komponentus ir apibrėžia sąveiką tarp jų.

MVC šablono sudaro trys elementai:

1. Valdiklis (angl. Controller) – siunčia komandas vaizdui (angl. View) ir modeliui (angl. Model). Čia negali būti atliekama jokia biznio logika, o parametrai gaunami per HTTP užklausą.
2. Vaizdas – Gauna reikiamą atvaizduoti informaciją iš valdiklio ir atvaizduoja suformuotą HTML dokumentą. Parametrai negali būti gaunami per HTTP užklausą, taip pat čia negali būti vykdoma jokia biznio logika.
3. Modelis – atlieka visą biznio logiką, čia parametrai negali būti gaunami per HTTP užklausą.

Visi šie elementai sąveikauja tarpusavyje ir sukuria darnią sistemą leidžiančią sukurti stabilią ir našią internetinę svetainę.

Vizualiai atvaizduotą MVC schemą galima pamatyti žemiau pateiktame paveikslėlyje (Pav. 1).



Pav. 1: MVC šablono grafinis vaizdas

Dauguma dabartinių karkasų naudoja MVC šablono. Taip išlieka aiški programinės įrangos struktūra ir keičiantis programuotojams kuriant tam tikrą programinę įrangą yra lengviau suprasti kitų programuotojų rašytą kodą.

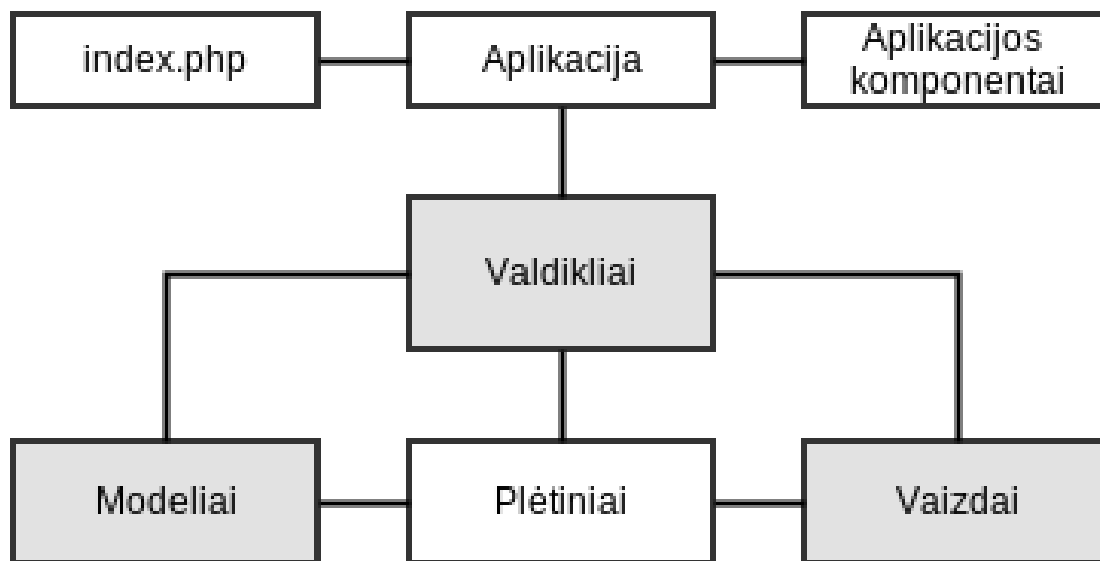
Toliau nagrinėjami trys pagrindiniai PHP karkasai internetinių svetainių kūrimui naudojantys MVC šabloną.

2.2 Yii karkaso apžvalga

Yii projektas buvo pradėtas 2008 metais, sausio 1 dieną ir buvo skirtas ištaisyti PRADO karkaso trūkumus. Pradinė PRADO versija lėtai apdorojo sudėtingus web puslapius, buvo sudėtinga ir daugelį valdiklių pritaikyti buvo kėblu, o Yii tuo metu buvo daug lankstesnis. Po 10 mėnesių privataus kūrimo buvo išleista pirmoji alfa Yii versija, o Gruodžio 3 dieną, Yii 1.0 buvo oficialiai paleistas [2].

2.2.1 Yii architektūrinis vaizdas

Yii architektūra paremta MVC šablonu ir jos pagrindą sudaro modeliai, valdikliai ir vaizdas. Visas Yii karkaso architektūrinis vaizdas pateiktas žemiau esančiame paveiksle (Pav. 2).



Pav. 2: Yii karkaso architektūrinis vaizdas

Visas aplikacijos veikimas vykdomas per index.php failą. Aplikacijos architektūroje taip pat yra atskirti komponentai, bei plėtiniai. Plėtiniai paprastai yra sukurti trečiųjų šalių ir yra publikuojami karkaso tinklalapyje.

2.2.2 Yii karkaso įrankiai

Yii karkasas turi daug papildomų pranašumų suteikiančių įrankių iš kurių svarbiausi yra šie:

- Gii – Yii modulis skirtas kodo generavimui. Šio modulio pagalba galima labai greitai sukurti valdiklius, bei visas CRUD operacijas.
- CGridView- vienas iš Yii įrankių leidžiantis kurti dažniausiai naudojamus komponentus darbui su lentelėmis. Yra kuriami: filtrai, rūšiavimai, puslapiavimas rašant minimaliai kodo. Tereikia paduoti duomenų perdavimo objektą, kuriam bus priskirti visi reikiami duomenys.
- jQuery- tai viena populiariausių JavaScript bibliotekų. Yii karkasas bei jo plėtiniai ir komponentai yra suintegruoti su jQuery biblioteka, tad visus veiksmus galima efektyviai kontroliuoti kliento pusėje naudojantis šia biblioteka.

2.2.3 Yii Karkaso trūkumai

Be visų išvardintų privalumų, Yii turi ir trūkumų, pagrindinis jų - Yii automatizuotos funkcijos. Yii sukuria CRUD operacijas iš vienos duomenų bazės lentelės, todėl atsižvelgiant į reikalavimus, sukurti formą ar duomenų tinklus iš dviejų modelių gali būti sudėtinga. Taip pat naudojant Cgrid vaizdą kai kurios modifikacijos gali reikalauti klasių ar karkaso perrašymo.

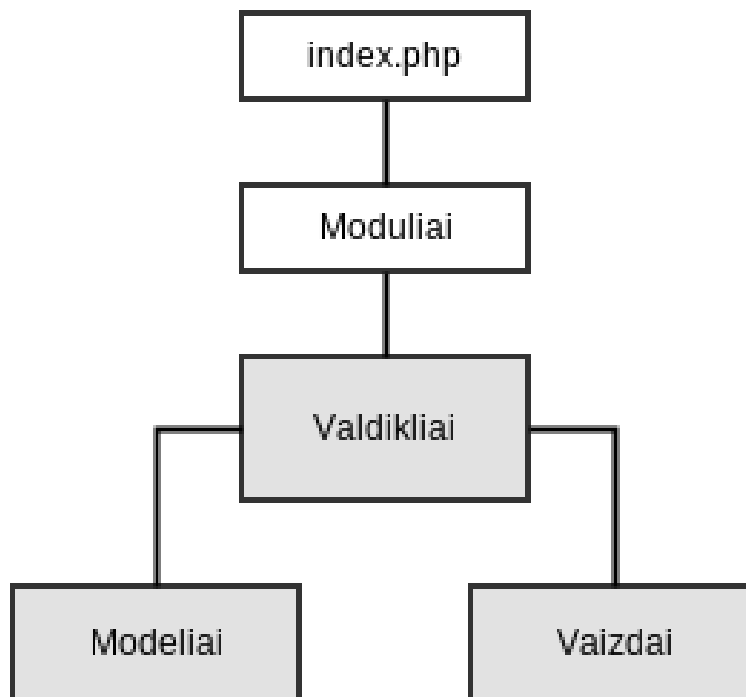
Dabartinė Yii karkaso versija yra 1.1.13, ir lyginant su Zend 2 ir Symfony2, kurie yra gana neseniai pilnai perrašyti, pastaroji yra gana sena, tačiau šiuo metu jau yra paleista viešai prieinama Yii karkaso antroji versija beta stadijoje. [2]

2.3 Zend 2 karkaso apžvalga

Neseniai išleistame Zend 2 karkase yra beveik visos pažangiausios PHP funkcijos. Šis karkasas turi labai mažai panašumų su savo pirmtaku Zend 1 karkasu. Zend 2 karkasas tinka sudėtingų aplikacijų kūrimui ir idealiai veikia su ORM doktrina, todėl tinkamai suplanuoti darbai gali duoti puikių rezultatų[3].

2.3.1 Zend 2 karkaso architektūrinis vaizdas

Zend 2 architektūra paremta MVC šablonu ir ją papildo moduliai. Visas karkaso architektūrinis vaizdas pateiktas žemiau esančiame paveiksle (Pav. 3).



Pav. 3: Zend 2 karkaso architektūrinis vaizdas

Sistemos darbas pradedamas vykdyti per index.php failą, čia yra kraunami moduliai. Kiekvienas modulis karkase turi savo atskirą MVC struktūrą. Moduliai yra viešinami Zend karkaso svetainėje, iš kur galima parsisiųsti jau sukurtų modulių atliekančių tam tikras funkcijas.

2.3.2 Zend 2 karkaso ypatybės

Zend 2 akcentuoja sistemų logikos dalinimą į kelis modulius, o ne valdiklius. Kiekvienas modulis sistemoje veikia kaip atskira sistema, su atskira konfigūracija ir išdėstymu. Toks būdas ne tik pagerina sistemos darbą bet ir yra lengvai redaguojamas, bei naudojant tokią struktūrą klaidos yra žymiai lengviau aptinkamos[3].

Zend 2 karkase gausu papildomų įrankių padedančių greičiau ar paprasčiau kurti sistemas. Išskirtiniai Zend 2 įrankiai yra šie:

- Servisų valdiklis - tai sluoksnis, kuris ne tik leis naudoti bet kokio serviso objektą savo aplikacijoje, bet ir leis jį redaguoti ir pritaikyti.
- PHP composer - tai įrankis užtikrinantis, kad visi projektai naudotų tas pačias versijas ir bibliotekas, kitaip sakant, tai priklausomumo valdymo įrankis.

2.3.3 Zend 2 karkaso trūkumai

Zend 2 karkaso dokumentacija tęsia Zend1 karkaso tradicijas. Susidūrus su klaidomis ar sunkumais, lengviau rasti reikiamą informaciją neoficialiuose bloguose, nei oficialioje dokumentacijoje.

Taip pat Zend karkasas turi sudėtingą direktorijų struktūrą, kurią reikės sukurti rankiniu būdu, aplanką po aplanko.

Dar vienas Zend 2 karkaso trūkumas yra tai, kad jis neturi failų generatoriaus, kuris paprastai būną įdiegtas į kitus karkasus.

2.4 Symfony2 karkaso apžvalga

Symfony2 karkasas yra sukurtas SensioLabs kompanijos, kuri turi daugiau nei 13 metų patirtį programinės įrangos kūrimo ir aktyvią tarptautinę bendruomenę. Symfony taip pat yra lengvai pritaikomas ir leidžia pasirinkti skirtingus programinės įrangos komponentus naudojamus projekte[4].

2.4.1 Symfony2 karkaso paketai

Symfony2 paketai (angl. Bundles) – tai lyg komponentai, kurie turi savo failų struktūrą. Sistemoje kurtoje pasitelkiant Symfony2 karkasą beveik viskas yra sudaryta iš paketų. Symfony2 paketai yra viešinami jų puslapyje, tad norint kurti naują sistemą, tereikia parsisiųsti ir įsidiegti juos į savo sistemą.[4]

2.4.2 Symfony2 karkaso įrankiai

Kaip ir į Zend 2 karkasą taip ir į Symfony2 karkasą yra integruotas PHP composer. Tai priklausomumo valdymo įrankis užtikrinantis, kad visi projektai naudotų tas pačias versijas ir bibliotekas.

Taip pat į šį karkasą yra integruotas Doctrine - tai PHP objektų sąryšių apjungimo karkasas. Vienas dažniausių uždavinių kuriant sistemą yra duomenų įvedimas ir išgavimas iš duomenų bazės. Doctrine įrankis padeda atlikti šiuos veiksmus symfony 2 karkase.

Symfony2 karkasas naudoja Twig kaip šablonų variklį. Verta paminėti jog šis šablonų variklis yra sukurtas tos pašios kompanijos - SensioLabs.

Į Symfony2 karkasą yra integruotas kodo generatorius. Norint sugeneruotą Symfony paketą (angl. Bundle) tereikia paleisti reikiamą komandą ir paketas būna sugeneruotas. Deja lyginant su Yii

karkasu CRUD generatoriaus veikiamas čia šiek tiek skiriasi. Yii karkaso atveju tereikia paleisti per naršyklę Gii modulį ir paspausti vieną kartą, kad sugeneruotų visus modelius, o norint sugeneruoti duomenų bazės subjektus (angl. Entity) Symfony2 reikia po vieną komandą leisti komandinėje eilutėje ir patvirtinti įvairius nustatymus, kol sugeneruojamas vienas DB subjektas.[4]

2.4.3 Symfony2 karkaso trūkumai

Symfony2 karkasas naudoja tokius įrankius kaip Doctrine, Twig, todėl jei programuotojas nėra dirbęs su šiais įrankiais, jam bus gana sunku perprasti karkaso veikimą. Taip pat tas pats kodo generatorius nėra toks paprastas koks galėtų būti, todėl galima daryti išvadą, jog Symfony2 karkasas yra vienas iš sudėtingiausių ir ilgiausiai užtrunkančių išmokti.

3 QR KODŲ GENERAVIMO IR STATISTIKŲ PATEIKIMO SISTEMOS KŪRIMAS, PANAUDOJANT SKIRTINGUS KARKASUS, PROJEKTYNĖ DALIS

Norint iširti karkasus suprojektuojama sistema, kuri vėliau yra sukuriama su trimis skirtingais PHP karkasais.

Sistemos paskirtis generuoti QR kodą, bei leisti QR kodo kūrėjui sekti vartotojų lankomumo statistiką.

3.1 QR kodų generavimo ir statistikų pateikimo sistemos tikslai

Sistema bus skirta visoms operacinėms sistemoms, kurios naudoja populiariausias naršyklės, taip pat ir mobiliems įrenginiams.

Įrankis privalo turėti šias galimybes:

- Vartotojo registravimosi galimybę;
- Vartotojo prisijungimo prie sistemos galimybę;
- Aplikacijos kūrimas;
- QR kodo spausdinimas;
- Lankomumo statistikų peržiūra.

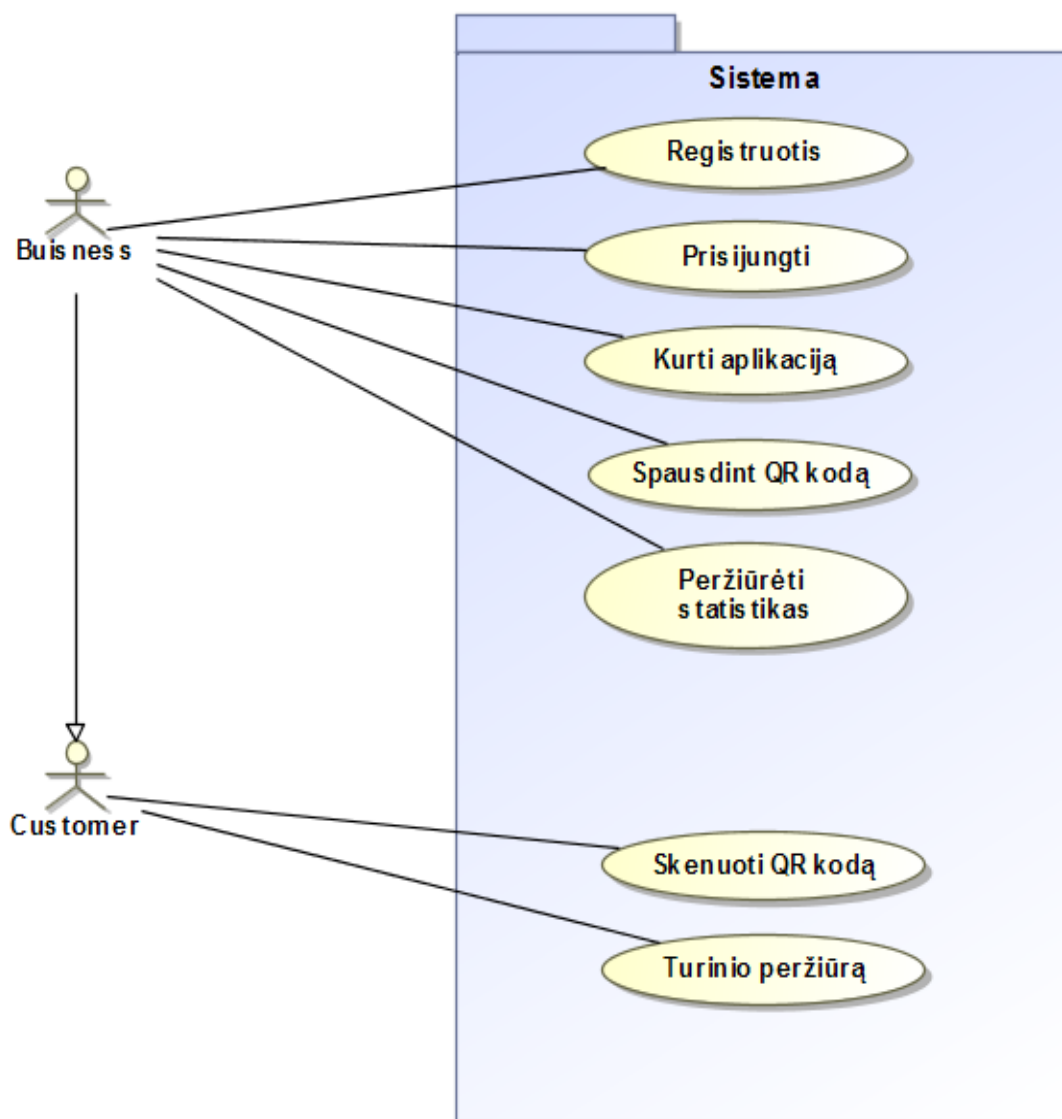
3.2 Bendri apribojimai

Kuriama sistema veiks WEB pagrindu, todėl vartotojo kompiuteryje privalės būti įdiegta naršyklė. Sistemos vartotojo klientai turės turėti mobiliuosius įrenginius su QR skaitymo programine įranga.

Duomenų saugojimui bus naudojama MySQL duomenų bazė. QR kodų atvaizdai bus saugomi serveryje.

3.3 Sistemos panaudojimų atvejų vaizdas

Žemiau pateiktame paveiksle (Pav. 4) yra pateiktas QR kodų generavimo ir vartotojų lankomumo sekimo PA vaizdas. Toliau pateiktose lentelėse aprašytas kiekvienas PA atvejis.



Pav. 4: Panaudojimo atvejų atvaizdas

Lentelė 1: PA „Registruotis”

Eil. Nr.	1
Panaudos atvejis	Registruotis
Tikslas	Leisti vartotojui registruotis prie sistemos
Aktoriai	Business
Ryšys su kitais PA	Nėra
Nefunkciniai reikalavimai	Vartotojo įvedama informacija bus validuojama (el. paštas, slaptažodis ir kt..)
Prieš-sąlyga	Atidarytas registracijos langas
Sužadinimo sąlyga	Registruojami duomenys
Po sąlyga	Siunčiamas el. laiškas su patvirtinimu. Patvirtinus gaunamas pranešimas, jog registracija pavyko
Pagrindinis scenarijus	Atvaizduojamas registracijos langas; Suvedami duomenys ir spaudžiamas patvirtinimo mygtukas; Išsaugomi duomenys;
Alternatyvus scenarijus	Nėra

Lentelė 2: PA „Prisijungti”

Eil. Nr.	2
Panaudos atvejis	Prisijungti
Tikslas	Patekti vartotojui į valdymo skydą
Aktoriai	Business
Ryšys su kitais PA	Nėra
Nefunkciniai reikalavimai	Nėra
Prieš-sąlyga	Atidarytas langas
Sužadinimo sąlyga	Tikrinami duomenys
Po sąlyga	Atvaizduojami rezultatai: klaida, sėkmingo prisijungimo žinutė (sėkmingo prisijungimo metu duomenys bus saugomi į sesiją)
Pagrindinis scenarijus	Leisti vartotojui prisijungti
Alternatyvus scenarijus	Nėra

Lentelė 3: PA „Kurti aplikaciją”

Eil. Nr.	3
Panaudos atvejis	Kurti aplikaciją
Tikslas	Leisti vartotojui kurti aplikaciją
Aktoriai	Business
Ryšys su kitais PA	Prisijungus yra suteikiama galimybė kurti kelių tipų WEB aplikacijas, bei vėliau jas redaguoti.
Nefunkciniai reikalavimai	Nėra
Prieš-sąlyga	Iškviečiamas WEB aplikacijos kūrimo langas.
Sužadinimo sąlyga	Pasirenkama WEB aplikacija, pateikiamas turinys.
Po sąlyga	Sukuriama WEB aplikacija ir pateikiamas QR kodas, kurį galima parsisiųsti ar atspausdinti.
Pagrindinis scenarijus	Sukurti WEB aplikaciją
Alternatyvus scenarijus	Nėra

Lentelė 4: PA „Palyginti technologijas”

Eil. Nr.	4
Panaudos atvejis	Spausdinti QR kodą
Tikslas	Suteikti vartotojui galimybę atsispausdinti QR kodą
Aktoriai	Business
Ryšys su kitais PA	QR kodą generuoti bus suteikiama galimybė tik tada, kai WEB aplikacija bus sukurta.
Nefunkciniai reikalavimai	Nėra
Prieš-sąlyga	Pateikiami šablonai
Sužadinimo sąlyga	Naršyklė iškviečia spausdinimo funkciją
Po sąlyga	Nėra
Pagrindinis scenarijus	Atspausdinti QR kodą
Alternatyvus scenarijus	Nėra

Lentelė 5: PA „Peržiūrėti statistikas”

Eil. Nr.	5
Panaudos atvejis	Peržiūrėti statistikas
Tikslas	Suteikti vartotojui galimybę peržiūrėti jo WEB aplikacijos apsilankymo statistikas
Aktoriai	Business
Ryšys su kitais PA	Statistikas bus galima peržiūrėti prisijungus prie sistemos.
Nefunkciniai reikalavimai	Nėra
Prieš-sąlyga	Atvaizduojamas statistikų langas
Sužadinimo sąlyga	Pateikiamos statistikos
Po sąlyga	Nėra
Pagrindinis scenarijus	Atvaizduoti statistikas
Alternatyvus scenarijus	Nėra

Lentelė 6: PA „Skenuoti QR kodą”

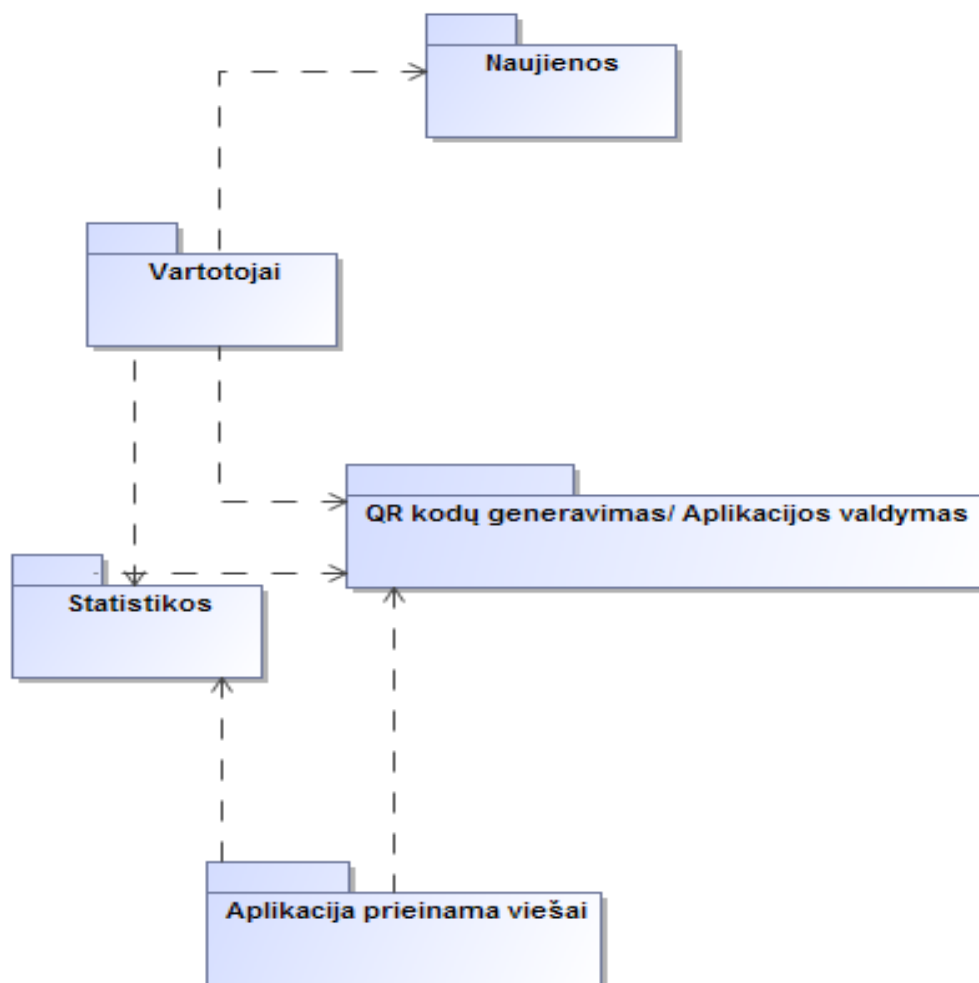
Eil. Nr.	6
Panaudos atvejis	Skenuoti QR kodą
Tikslas	B2C principu suteikiama biznio klientui nuskenuoti QR kodą ir pamatyti turinį.
Aktoriai	Client
Ryšys su kitais PA	Nėra
Nefunkciniai reikalavimai	Nėra
Prieš-sąlyga	Atvaizduojama nuoroda
Sužadinimo sąlyga	Paspaudus nuorodą, serveryje randama WEB aplikacija
Po sąlyga	Atvaizduojami duomenys
Pagrindinis scenarijus	Atvaizduoti informaciją
Alternatyvus scenarijus	Nėra

Lentelė 7: PA „Turinio peržiūra”

Eil. Nr.	7
Panaudos atvejis	Turinio peržiūra
Tikslas	Leisti vartotojui peržiūrėti turinį
Aktoriai	Client
Ryšys su kitais PA	Nuskenavus QR kodą gaunama nuoroda į turinio peržiūrą.
Nefunkciniai reikalavimai	Nėra
Prieš-sąlyga	Pateikiamas turinys
Sužadinimo sąlyga	Yra galimybė vykdyti tam tikrus veiksmus, priklausomai nuo WEB aplikacijos tipo.
Po sąlyga	Nėra
Pagrindinis scenarijus	Atvaizduoti turinį
Alternatyvus scenarijus	Nėra

3.4 Sistemos statinis vaizdas

Sistema suskaidyta į 5 paketus aukščiausiam lygį, kurie pavaizduoti 5 paveiksle.



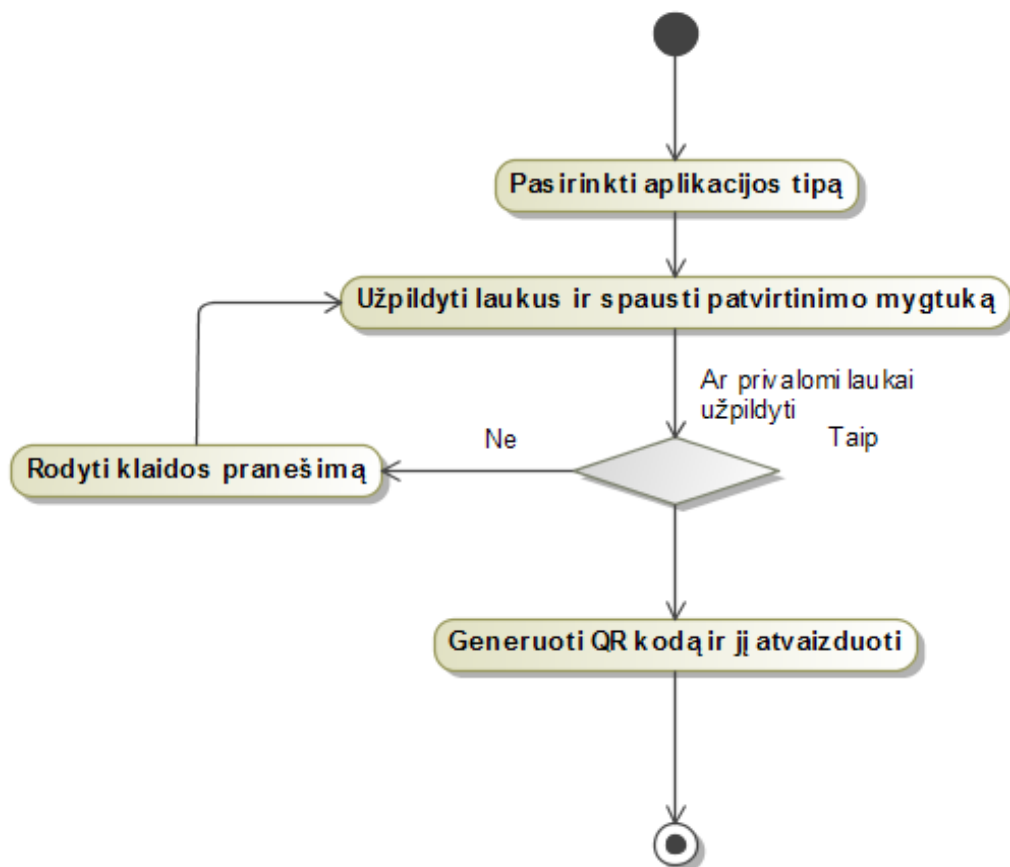
Pav. 5: Paketų diagrama

Pav. 5 matomi pagrindiniai paketai. Paketas „Naujienos” yra atsakingas už naujienų pateikimą. Naujienos yra pateikiamos blogo stiliumi. Paketas „Vartotojai” yra atsakingas už vartotojų teises, prisijungimus, bei registracijas. Statistikų paketas pateikia WEB aplikacijos lankomumo statistikas, o už pačių WEB aplikacijų kūrimą, trynimą ir redagavimą yra atsakingas paketas „QR kodų generavimas/ Aplikacijos valdymas”. Aplikacijos pateikimą vartotojui apdoroja paketas - „Aplikacija prieinama viešai”.

3.5 Sistemos veiklos diagramos

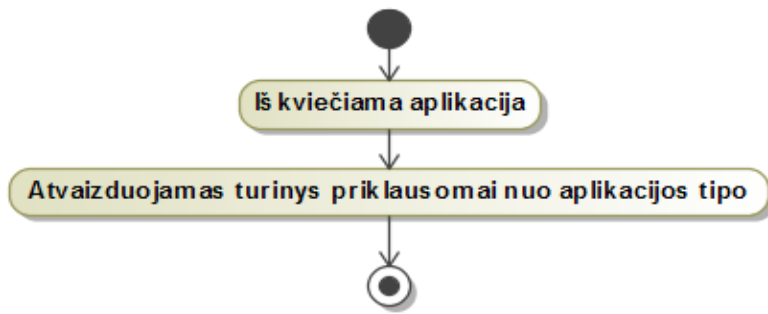
Veiklos diagrama yra viena iš dažniausiai naudojamų diagramų tipų skirtų atvaizduoti sistemos veikimo principą.

Toliau pateikiamos kuriamos sistemos veiklos diagramos (pav. 6, pav 7, pav 8.).



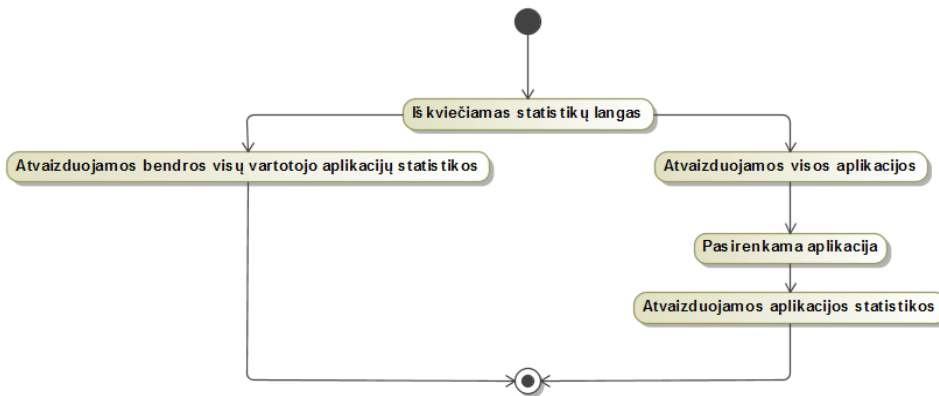
Pav. 6: QR kodų generavimas ir aplikacijų kūrimas

Pav. 6 pateikta veiklos diagrama- QR kodų generavimas ir WEB aplikacijų kūrimas. Čia pasirenkamas WEB aplikacijos tipas, tuomet atvaizduojamas langas kuriame reikia užpildyti laukus bei patvirtinti informaciją spaudžiant patvirtinimo elementą, jei privalomi užpildyti laukai nėra užpildyti, veiksmai privalo būti kartojami, kol privalomi laukai bus užpildyti. Užpildžius laukus yra generuojamas QR kodas ir WEB aplikacija. QR kodas yra atvaizduojamas.



Pav. 7: Aplikacijų pateikimas viešojoje dalyje

Pav. 7 atvaizduotas WEB aplikacijų pateikimas viešojoje dalyje. Yra iškviečiama WEB aplikacija, tuo metu registruojama vartotojo informacija. Tuomet yra pateikiamas turinys priklausomai nuo kliento įvestos informacijos.

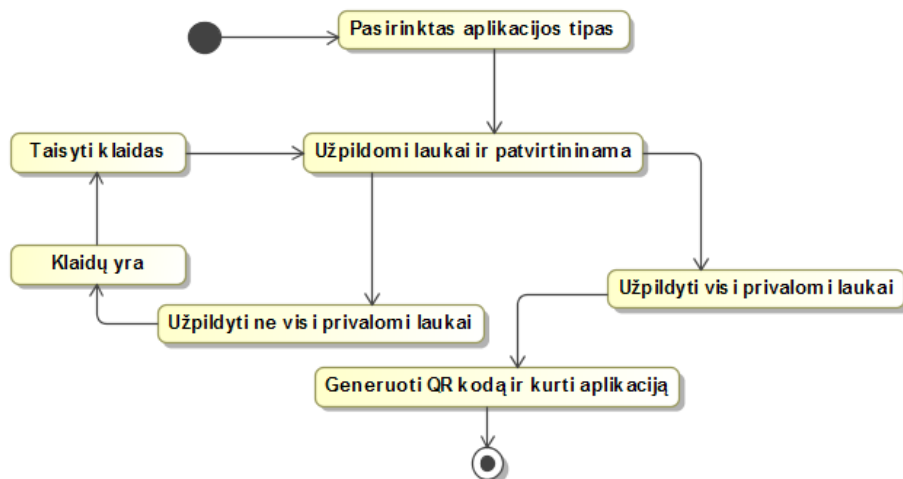


Pav. 8: Statistikų atvaizdavimas

Pav. 8 yra pavaizduotas WEB aplikacijų statistikų lango iškvietimas, kuriame yra atvaizduojamos bendros visų aplikacijų statistikos, taip pat yra atvaizduojamos ir visos aplikacijos, pasirinkus vieną iš jų yra matoma individualios aplikacijos vartotojų lankomumo statistika.

3.6 Sistemos būsenų diagramos

Sistemos būsenų diagramos atvaizduoja svarbiausias verslo ar veiklos sistemos būsenas, bei jų kitimą toje pačioje sistemoje. Toliau pateikiamos kuriamos sistemos būsenų diagramos.



Pav. 9: QR kodų generavimas ir WEB aplikacijų kūrimas

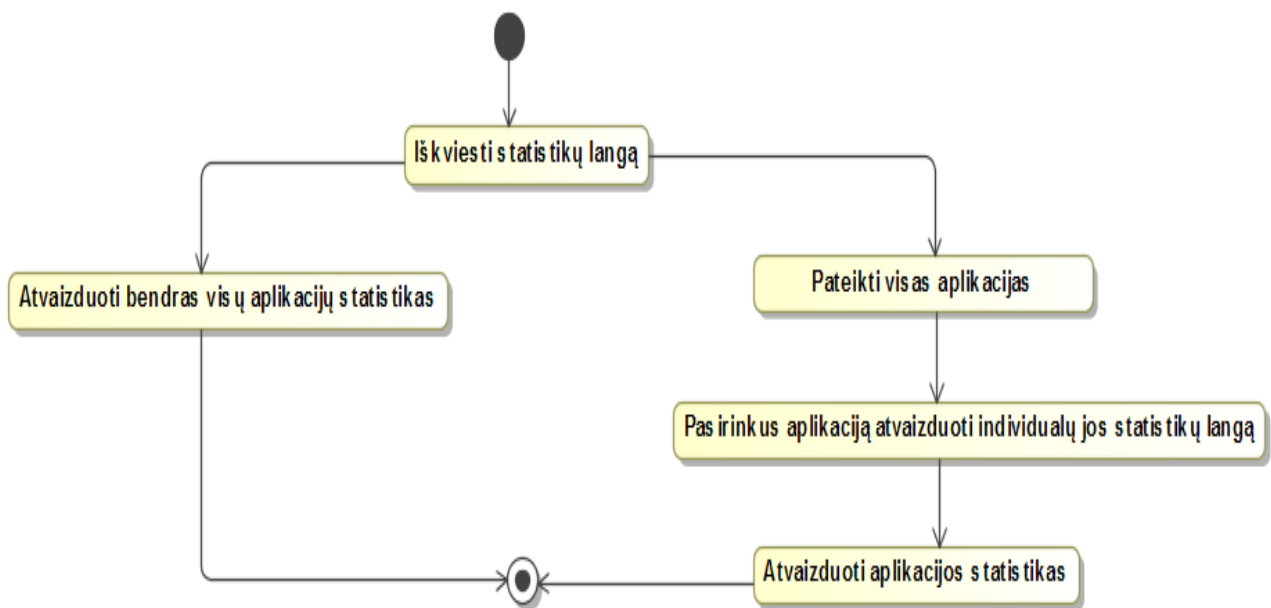
Pav. 9 būsenų diagrama atvaizduoja QR kodų generavimą ir WEB aplikacijų kūrimą. Yra pasirenkamas aplikacijos tipas, tuomet atvaizduojamas langas kuriame reikia užpildyti laukus bei patvirtinti informaciją spaudžiant patvirtinimo elementą, jei privalomi užpildyti laukai nėra užpildyti veiksmai privalo būti kartojami, kol privalomi laukai bus užpildyti. Užpildžius laukus yra generuojamas QR kodas ir aplikacija. QR kodas yra atvaizduojamas.



Pav. 10: Aplikacijų pateikimas viešojoje dalyje

Pav. 10 atvaizduojamas aplikacijų pateikimas viešojoje dalyje.

Iškviečiama aplikacija, tuo metu yra registruojama vartotojo informacija. Galiausiai yra pateikiamas turinys priklausomai nuo kliento įvestos informacijos duomenų bazėje.

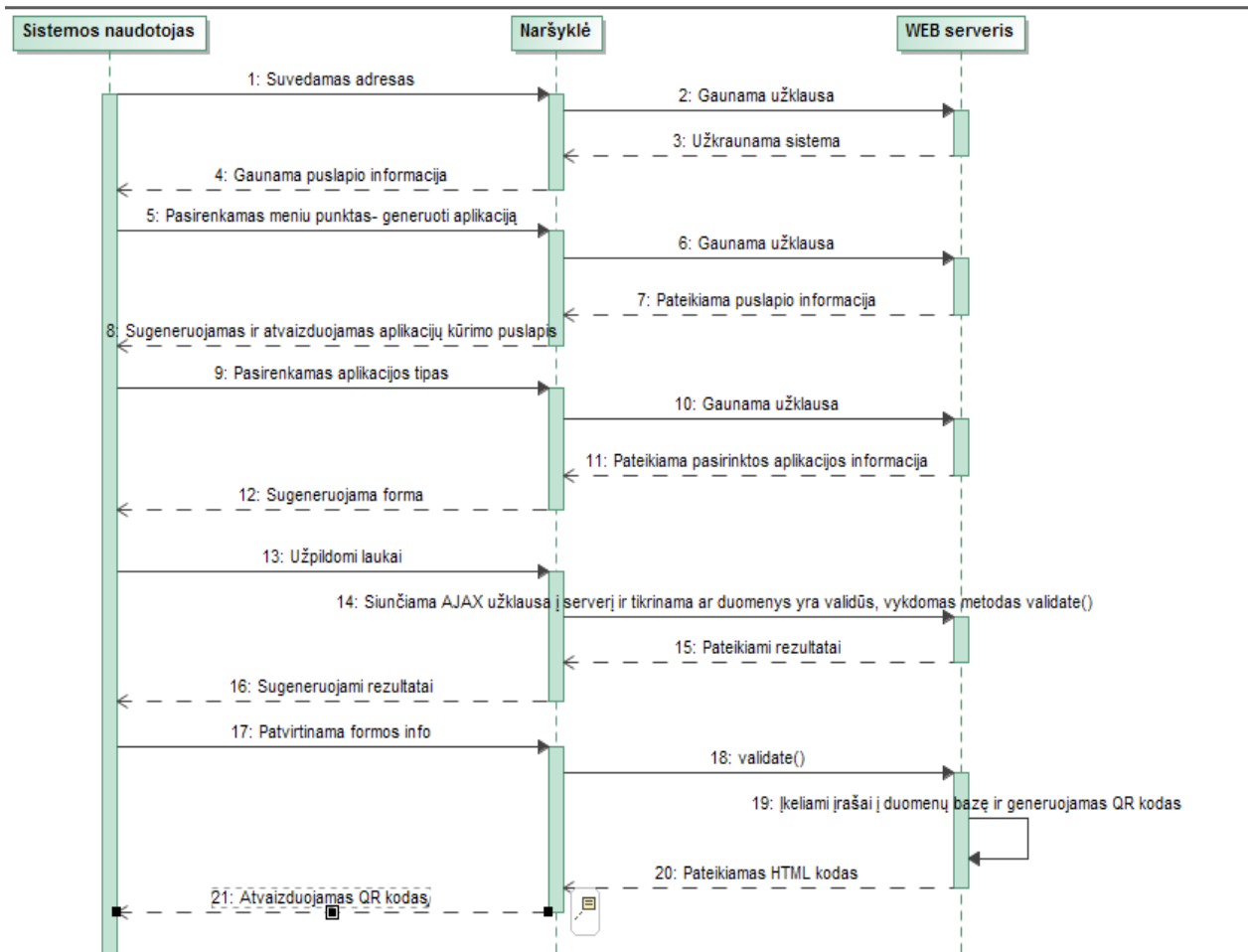


Pav. 11: Statistikų atvaizdavimas

Yra iškviečiamas statistikų langas kuriame yra atvaizduojamas bendros visų WEB aplikacijų statistikos, taip pat yra atvaizduojamos visos aplikacijos, pasirinkus vieną iš jų yra matoma individualios WEB aplikacijos vartotojų lankomumo statistika. (17 pav.)

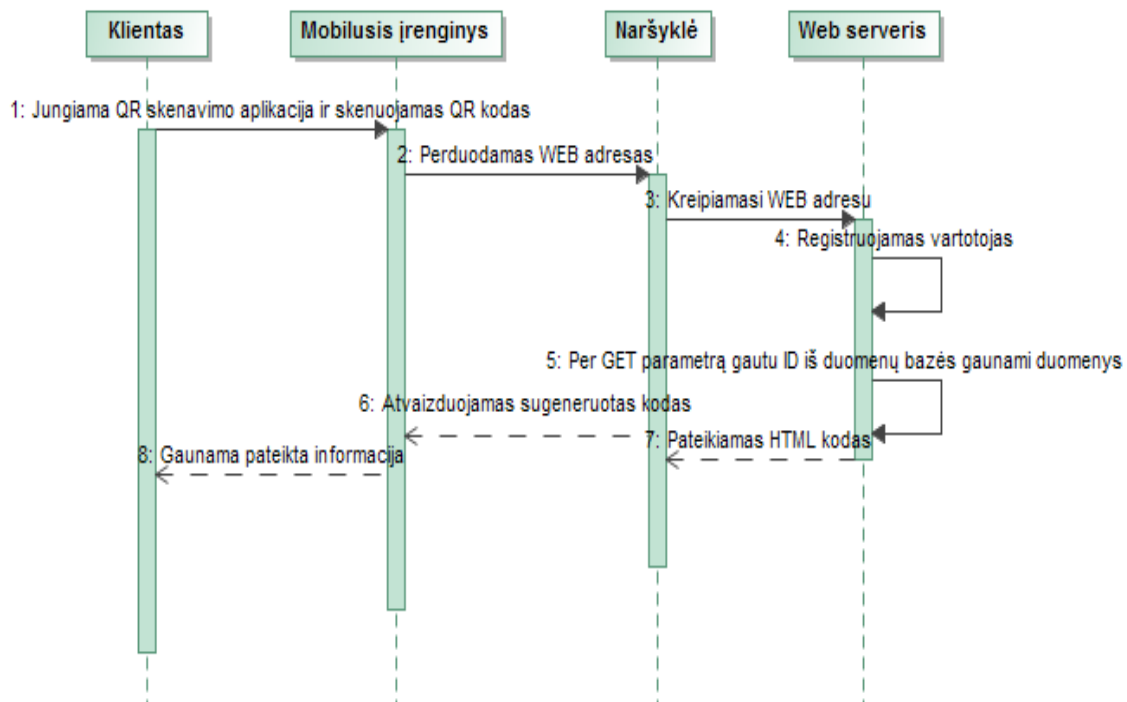
3.7 Sistemos sekų diagramos

Sistemos sekų diagramos iliustruoja objektų, jų būsenų, veiksmų lygiagretų išsidėstymą laike bei pranešimus tarp jų. Šiame poskyryje pateiktos sekų diagramos ir jų aprašymai.



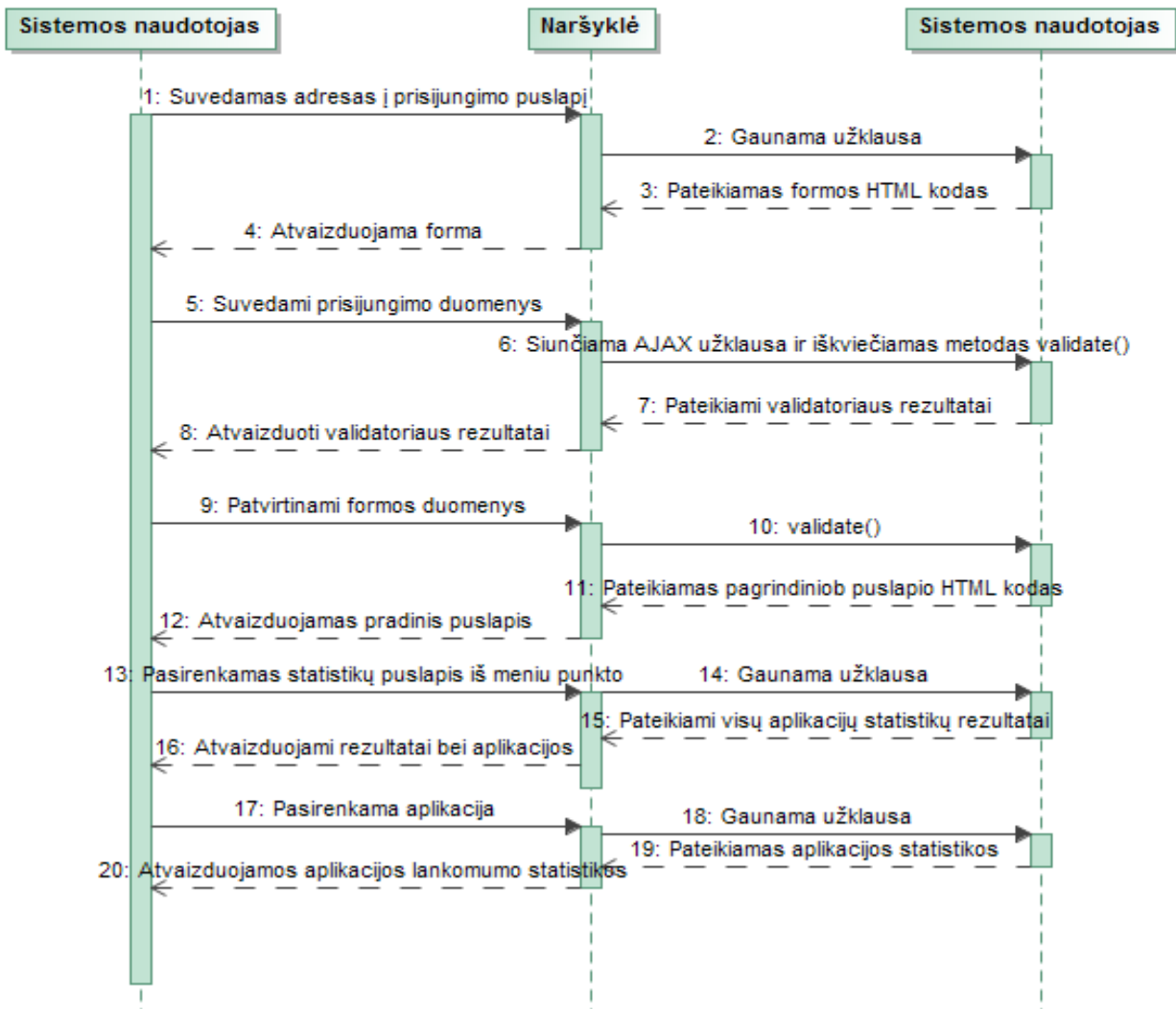
Pav. 12: QR kodų generavimas ir WEB aplikacijų kūrimas

Pav. 12 atvaizduota sekų diagrama- QR kodų generavimas ir WEB aplikacijų kūrimas. Čia patekus į aplikacijos kūrimo puslapį yra atvaizduojami aplikacijų tipai, pasirinkus vieną iš tipų, pagal jį atvaizduojama forma, kurią vartotojas turi užpildyti, užpildžius ir patvirtinus formą yra sugeneruojama aplikacija ir QR kodas.



Pav. 13: WEB aplikacijų pateikimas viešojoje dalyje

Pav. 13 atvaizduota sekų diagrama- aplikacijų pateikimas viešojoje dalyje. Vartotojas nuskenuoja QR kodą savo mobiliuoju įrenginiu, gautu web adresu keliauja į aplikaciją, čia patekęs yra užregistruojamas ir vartotojui pateikiamas turinys.

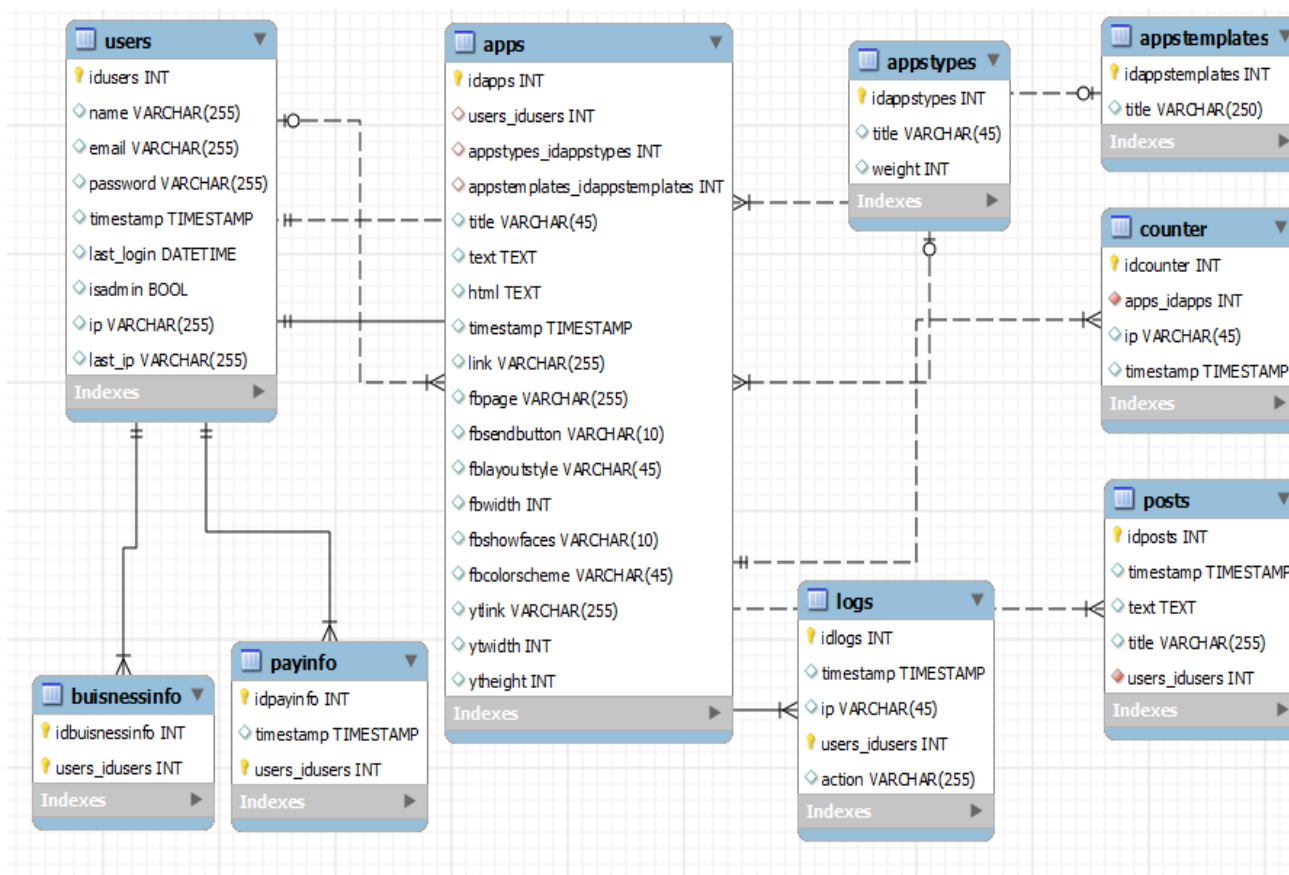


Pav. 14: Statistikų atvaizdavimas

Pav. 14 atvaizduota sekų diagrama - statistikų atvaizdavimas. Visų pirma sistemos naudotojas turi prisijungti prie sistemos, tą padaręs turi patekti į statistikų puslapį, ten yra atvaizduojamos visų aplikacijų statistikos bei visos aplikacijos. Pasirinkus vieną iš aplikacijų yra patenkama į individualų aplikacijos vartotojų lankomumo statistikų puslapį.

3.8 Sistemos duomenų vaizdas

Sistemos duomenų saugojimui naudojama MySQL duomenų bazė (Pav. 15).



Pav. 15: Duomenų bazės lentelė

Aplikacijų informacija bus saugoma *apps* lentelėje. Generuojami QR kodų paveikslėliai nebus saugomi duomenų bazėje. Jie bus talpinami FTP serveryje tam skirtoje direktorijoje, jų pavadinimas atitiks aplikacijų lentelėje esančius identifikatorius.

4 PHP KARKASŲ TYRIMO IR EKSPERIMENTINĖ DALIS

Renkantis tinkamą karkasą labiausiai trūksta informacijos apie šias 4 charakteristikas:

- Darbo imlumą
- Našumą
- Funkcionalumą
- Karkaso vidinę kokybę

Palyginus karkasus pagal šias keturias charakteristikas, būtų galima lengviau apsispręsti kokį karkasą rinktis naujam projektui.

4.1 Tyrimuose naudojamos sistemos parametrai

Vykdamas karkasų našumo tyrimą buvo naudojama tokia programinė ir aparatūrinė įranga:

- Procesorius: Intel Core i5-2450M CPU @ 2.50GHz × 4
- Darbinė atmintis: 7.7 Gb
- Operacinė sistema: Ubuntu 13.04 (64bit)
- Serveris: LAMP
- PHP versija: 5.4

4.2 PHP karkasų darbo imlumo tyrimas

Tyrimas buvo vykdomas pirmą kartą kuriant QR kodų generavimo ir statistikų pateikimo sistemą, buvo pasirinktas Yii karkasas. Vėliau sistema perdaryta Symfony2 karkasu, o dar vėliau Zend 2 karkasu.

5.1.1, 5.1.2 ir 5.1.3 poskyriuose aprašomas kūrimo procesas kuriant sistemą kiekvienu iš karkasų. Verta paminėti, jog prieš pradėdant programavimo darbus MySQL duomenų bazės schema buvo jau sugeneruota.

4.2.1 Yii karkaso darbo imlumo tyrimas

Prieš pradėdant programavimo darbus buvo vykdomas karkaso diegimas. Sunkumų nekilo, tereikėjo parsisiųsti karkasą iš oficialios karkaso svetainės. Čia panaudojus struktūros generavimo įrankį buvo sugeneruota pradinė aplikacija.

Pasinaudojus Yii moduliu, buvo sukuriami visi reikalingi valdikliai, taip pat ir visos CRUD operacijos. Viską atlikti buvo labai paprasta, reikėjo tik nurodyti duomenų bazės vardą ir buvo sugeneruotos visos CRUD operacijos visos lentelėms.

Nors šį kodą gana stipriai reikia modifikuoti, tačiau turint gerą pavyzdį, aplikacijos kūrimo darbai vyksta žymiai greičiau, nei neturint nieko. Taip pat šiame sugeneruotame kode buvo ne tik pagrindinės CRUD operacijos, bet ir tokios funkcijos kaip puslapiavimas, laukų validacija. Karkasas pilnai suintegruotas su jQuery biblioteka, norint jog sistema vykdytų klaidų tikrinimą neperkrovus puslapio, tereikia įskiepio konfigūracinėje eilutėje nurodyti teigiamą loginę reikšmę.

Vykdamas QR kodų generavimo ir lankomumo sekimo sistemos darbus sunkiausia buvo modifikuoti jau sugeneruotą kodą. Taip pat yra sudėtinga apjungti duomenų bazės lenteles ir naudotis jau su sugeneruotu kodu.

Sistemai didėjant modeliai tapo vis sudėtingesni ir kodo eilučių kiekis gana greitai išaugo. Galima numanyti, kad kuriant didesnę sistemą ir ją palaikant ilgesnį laiko tarpą modeliai gali tapti milžiniški ir juos suprasti gali būti labai sudėtinga.

4.2.2 Zend 2 karkaso darbo imlumo tyrimas

Prieš pradėdant kurti sistemą, karkasą buvo galima parsisiųsti vientisą, bei su esama struktūra. Deja Zend 2 karkasas neturi struktūros generatoriaus, tad tenka naudotis Git versijų kontrolės sistema, ir parsisiųsti visą karkasą kartu su esama struktūra.

Karkaso struktūra atrodo lengvai suprantama. Kiekvienas modulis turi savo atskirą MVC struktūrą, tad yra labai paprasta suprasti kodą, bei kurti naujus modulius. Deja karkasas neturi ne tik struktūros bet ir visiškai jokio kito generatoriaus, todėl tenka visus modelius, visas CRUD operacijas aprašyti ranka, tai užtrunka gana ilgą laiko tarpą. Karkase taip pat nėra jau suintegruotų formų validatorių, kurie kreiptųsi į PHP serverį per AJAX užklausas.

Nežiūrint į sunkią pradžią kuriant sistemą vėliau esamoje sistemoje susigaudyti buvo gana paprasta. Kadangi QR kodų generavimo ir statistikų sistemą išskaidžiau į atskirus modulius, tereikėjo tik konfigūraciniame faile aprašyti kuriuos modelius naudoti norimuose moduluose ir jie buvo pasiekiami. Lyginant Zend 2 karkaso architektūrą su Yii karkaso architektūra, jaučiamas ryškus skirtumas Zend 2 karkaso naudai.

4.2.3 Symfony2 karkaso darbo imlumo tyrimas

Karkasas yra parsiumčiamas naudojantis Composer įrankį. Papildomai reikia pakeisti kešo, bei registro direktorių teises ir karkasas jau buvo užkraunamas naršyklėje.

Karkaso struktūra yra šiek tiek modifikuota MVC, tačiau vis tiek lengvai suprantama. Nors lyginant su Zend 2 karkaso struktūra, ši atrodo kiek sudėtingesnė, kadangi sistemai augant paketo struktūros dydis taip pat plečiasi.

Darbas pradėdamas pasileidus generatorių ir sukuriant pirmąjį paketą. Paketo kelias nurodomas AppKernel.php faile. Pakete sukuriami Twig dokumentai, bei įterpiamas stilius. Taip pat papildomai į konfigūracinį failą nurodomi keliai.

Bandant generuoti visą kodą iš esamos duomenų bazės schemos kilo problemų dėl sąryšių, tad teko visus lentelių laukus aprašyti ranka sukurtose klasėse. Buvo aprašomi klasės kintamieji ir jų komentarai Doctrine įrankiui tinkamu formatu. Tai padarius buvo generuojamos nustatymo funkcijos (angl. Setters) ir gavimo funkcijos (angl. Getters). Taip pat panaudojus dar vieną papildomą komandą buvo sukurtos lentelės nurodytoje duomenų bazėje. Padarius pakeitimus tereikėjo nurodyti tam tikrą komandą ir duomenų bazė atsinaujino, tad papildomai jungtis prie DBVS net nereikėjo. Duomenų bazės migravimo skriptai, formos, bei CRUD operacijos kuriamos taip pat lengvai.

Pradėti naudotis šiuo karkasu yra gana sudėtinga. Neužtenka išmokti karkaso struktūros, reikia taip pat gerai žinoti ir su karkasu naudojamų įrankių galimybes, tačiau visą tai įvaldžius galima pasiekti labai gerų rezultatų.

4.3 PHP karkasų našumo tyrimas

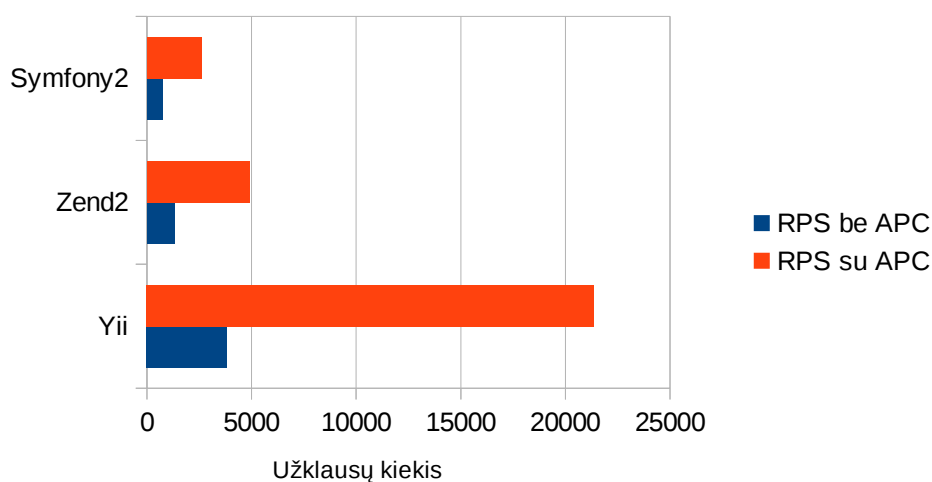
Norint palyginti karkasus reikia taip pat ištirti jų našumą. Tyrimo rezultatuose tikimasi pamatyti pranašiausią karkasą.

Tiriant karkasų našumą yra pasirenkamas „Sveikas pasauli“ testas, šio testo metu valdiklyje yra priskiriama reikšmė ir ši reikšmė perduodama atvaizduoti. Šio testo atlikimui buvo naudojama programa- *ApacheBench*. Ši programa pateikia tokius rezultatus kaip kreipinių kiekis per tam tikrą laiko tarpą.

Testas buvo vykdomas kiekvienam karkasui po 10 kartų be APC ir 10 kartų su APC. Visi testai buvo vykdomi po 30 sekundžių pasirinkus 10-tą užimtumo lygį, tai reiškia, kad vienu metu yra vykdoma 10 kitų užklausų.

4.3.1 Našumo tyrimo rezultatai

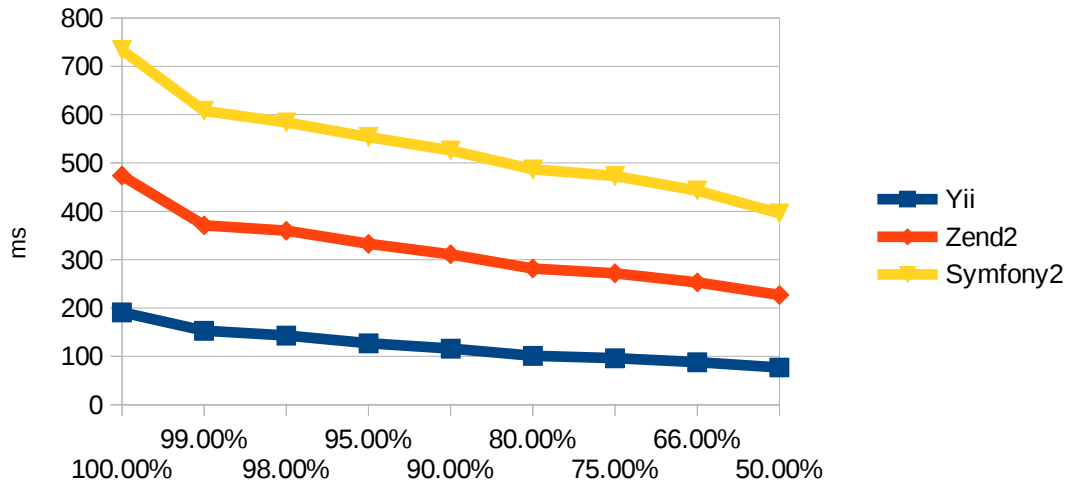
Tyrimo rezultatai pateikiami Pav. 16 ir Pav. 17



Pav. 16: Užklausų kiekis per 30 sekundžių

Pav. 16 matoma, jog Zend 2 karkasas šiame tyrime maždaug du kartus lenkia Symfony2 karkasą t. y. Symfony2 karkasas vidutiniškai atliko 750 užklausų per 30 sekundžių, o Zend 2 karkasas atliko 1320 užklausų, tačiau Yii karkasas buvo maždaug 3 kartus greitesnis už Zend 2 karkasą, jis vidutiniškai atliko 3800 užklausų per 30 sekundžių.

Ijungus APC, Yii karkaso aplikacija vidutiniškai atliko 21360, Zend 2- 4900, o Symfony2 tik 2630 užklausų per 30 sekundžių.



Pav. 17: Procentaliai pateiktas užklausų aptarnavimas per tam tikrą laiką

Pav. 17 pateiktas procentaliai atvaizduotas laikas. Matoma, jog Yii karkasas gali aptarnauti 100% užklausų per 190ms ir 50% užklausų per 80ms. Toliau seka Zend 2 karkasas ir labiausiai nuvilia Symfony2, kuris 100% užklausų aptarnauja tik per 740ms.

4.4 Karkasų funkcionalumo tyrimas

Pagal nutylėjimą įdiegti karkasai su tam tikra struktūra turi jau įdiegtus modulius ar tam tikrą funkcionalumą. Lentelėje nr. 8 yra pateikti tam tikri kriterijai, pagal kuriuos dalinai galima spręsti kokį karkasą rinktis tam tikram projektui.

Lentelė 8: Karkasų funkcionalumo skirtumai ir panašumai

	Yii	Zend 2	Symfony2
Dydis	20.9Mb	22..6Mb	17Mb
PHP versija	Nuo 5.1	Nuo 5.3.3	Nuo 5.3.2
MVC palaikymas moduluose	Yra	Yra	Yra
Failų DB	Yra	Ne	Ne
Šablonų variklis	Nėra	Nėra	Twig
JavaScript biblioteka naudojama pagal nutylėjimą	jQuery	jQuery	Nėra
CSS karkasas naudojamas pagal nutylėjimą	Blueprint	Twitter Bootstrap	yui
ORM	Active Record	SQL	Doctrine
Composer	Ne	Taip	Taip
Plėtinių/Modulių/Paketų kiekis viešinamų internetinėje erdvėje	1230	260	2005

Lentelėje nr. 8 matoma, jog Yii karkasas palaikomas nuo PHP 5.1 versijos. Lyginant su kitais karkasais ši versija gana sena, tai yra todėl, jog kiti karkasai buvo pilnai perrašyti ir buvo išleistos jų

antrosios versijos. Visi karkasai naudoja MVC struktūra, galima būtų paminėti, jog Symfony2 karkasas yra šiek tiek modifikavęs modelių dalį, tačiau funkcionalumas išliko tas pats.

Kaip šablonų variklį Symfony2 karkasas naudoja Twig, kiti karkasai nenaudoja jokio.

Tiek Yii karkasas, tiek ir Zend 2 karkasas pagal nutylėjimą naudoja jQuery biblioteką, skirtumas toks, jog Yii karkasas ne tik ją įtraukė į failų sąrašą, bet ir naudoja daugelyje sistemos komponentų.

Pagal nutylėjimą CSS karkasus visi PHP karkasai naudoja skirtingus. Stabilioji Yii karkaso versija naudoja Blueprint, tačiau dabar kuriamoje Yii2 jau bus diegiamas Twitter bootstrap CSS karkasas.

Kaip ORM Yii karkasas naudoja įrankį vadinamą- Active Record (angl. Aktyvus įrašas), didžiausias jo plusas yra greitis Yii karkase. Symfony2 naudoja trečiųjų šalių sukurtą įrankį Doctrine. Zend 2 karkasas nenaudoja jokio ORM įrankio.

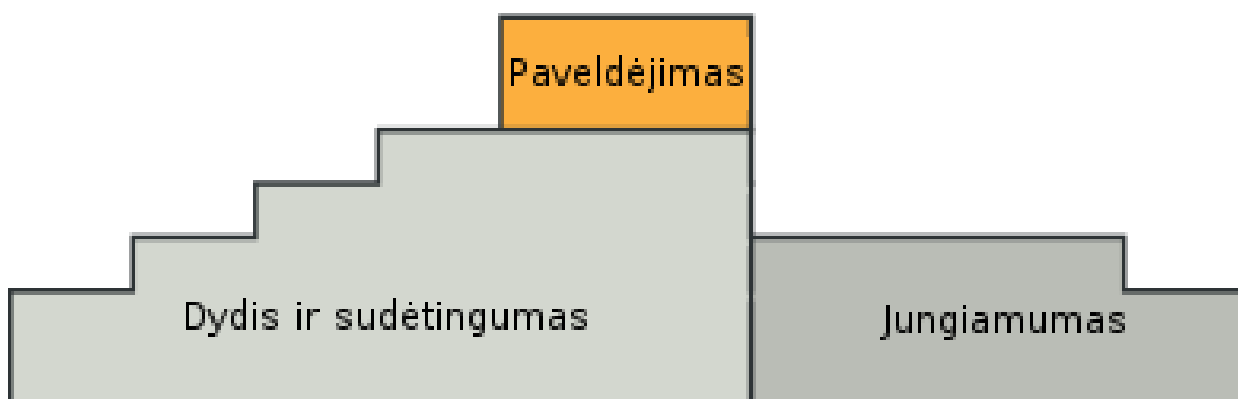
Kiekvienas iš karkasų turi pakankamai platų funkcionalumą ir ko nors pritrūkus karkase, galima parsisiųsti papildomą modulį, plėtinį ar paketą, todėl galima sakyti, jog didžiausia funkcionalumo dalis priklauso nuo karkaso bendruomenės, kuri ir kuria įvairius plėtinius, paketus ir modulius. Kol kas didžiausią bendruomenę turi Symfony2 karkasas, tad galima teigti, kad šis karkasas savo funkcionalumu lenkia likusius.

4.5 Karkasų kodo analizė

Pagrindinis analizės tikslas yra peržvelgti karkasų kodą ir įvertinti jo sudėtingumą.

Programinei įrangai apžvelgti yra naudojamas įrankis „PHP depend“ kuriuo analizuojamas programinės įrangos sudėtingumas.

Rezultatai yra pateikti apžvalgos piramidės diagrama, kuri skirta atvaizduoti programinės įrangos metrikoms. Apžvalgos piramidės struktūra yra pavaizduota Pav. 18



Pav. 18: Apžvalgos piramidės struktūra

4.5.1 Kodo analizės tyrimo aprašymas

Tyrimas vykdomas praleidžiant „PHP depend“ programą per kiekvienam iš karkasų. Taip yra gaunama trijų kodo metrikų kategorijų informacija:

- Dydžio ir sudėtingumo;
- Jungiamumo;
- Paveldėjimo.

Atsižvelgiant į šias metrikas galima nustatyti karkasų sudėtingumą, bei susidaryti pirmą įspūdį apie karkasą.

Kategorija- dydis ir sudėtingumas, nusako dažniausiai naudojamas programinės įrangos metrikas. Visos kategorijų metrikos pateiktos lentelėse nr. 1-3.

Lentelė 9: Kategorijos- dydis ir sudėtingumas, kodo metrikos

Metrika	Aprašymas
NOP	Paketų kiekis
NOC	Klasių kiekis
NOM	Metodų kiekis
LOC	Kodo eilučių kiekis
CYCLO	McCabe ciklomatinis sudėtingumas.

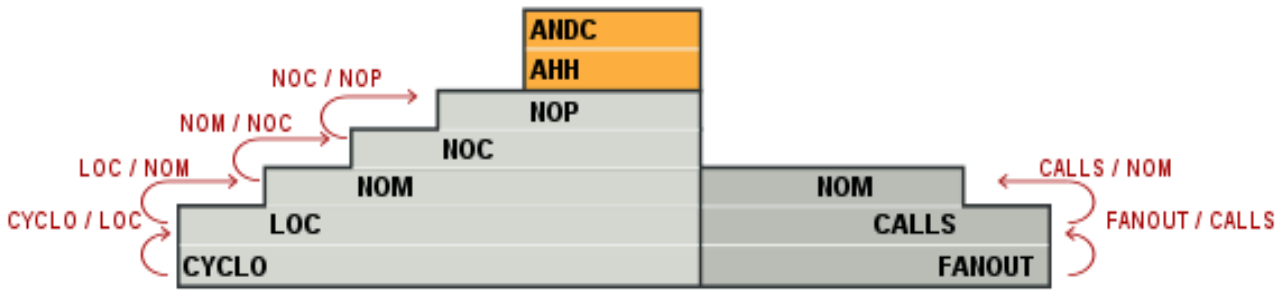
Lentelė 10: Kategorijos- jungiamumas, kodo metrikos

Metrika	Aprašymas
CALLS	Skaičius metoduose esančių kreipinių į kitus metodus.
FANOUT	Klasių ir sąsajų kiekis, kurios nėra toje pačioje paveldimumo šakoje.

Lentelė 11: Kategorijos- paveldėjimas, kodo metrikos

Metrika	Aprašymas
ANDC	Paveldimų klasių vidurkis.
AHH	Paveldėtų klasių vidutinis gylis. Dešimties klasių sistemoje, metrikos reikšmė 1 būtų tuo atveju, jei 5 klasės būtų paveldėtos kitų 5 klasių arba 5 klasės paveldėtų vieną šakninę klasę.

Peržiūros piramidėje skaičiavimai vykdomi iš apačios į viršų, dalinant apačioje esančią reikšmę iš aukščiau esančios reikšmės taip kaip pateiktas pav. 19.



Pav. 19: Skaičiuojamos reikšmės peržiūros piramidėje

Atsižvelgiant į rezultatus galima klasifikuoti rezultatus pagal pateiktas reikšmes lentelėje nr. 4.

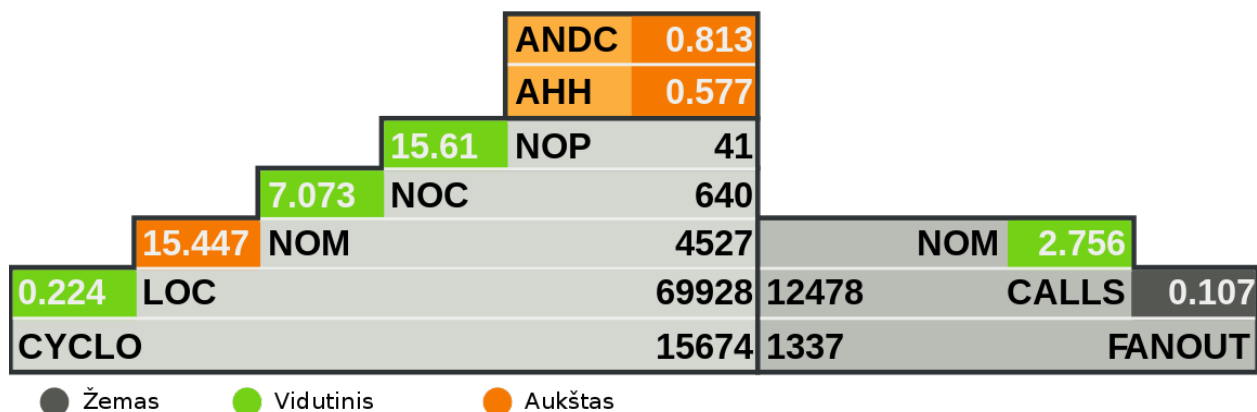
Lentelė 12: Metrikų atskaitinės vertės

Metrika	Žemas sudėtingumas	Vidutinis sudėtingumas	Aukštas sudėtingumas
CYCLO/LOC	0.16	0.20	0.24
LOC/NOM	7	10	13
NOM/NOC	4	7	10
NOC/NOP	6	17	26
CALLS/NOM	2.01	2.62	3.2
FANOUT/CALLS	0.56	0.62	0.68
ANDC	0.25	0.41	0.57
AHH	0.09	0.21	0.32

Tyrimo rezultatai matomi peržiūros piramidėse toliau esančiuose 4.5.2, 4.5.3 ir 4.5.4 poskyriuose[8].

4.5.2 Yii karkaso kodo analizės tyrimo rezultatai

Atlikus tyrimą buvo gauta apžvalgos piramidė, kuri pateikta pav. 20.



Pav. 20: Yii karkaso tyrimo rezultatai pateikti apžvalgos piramide

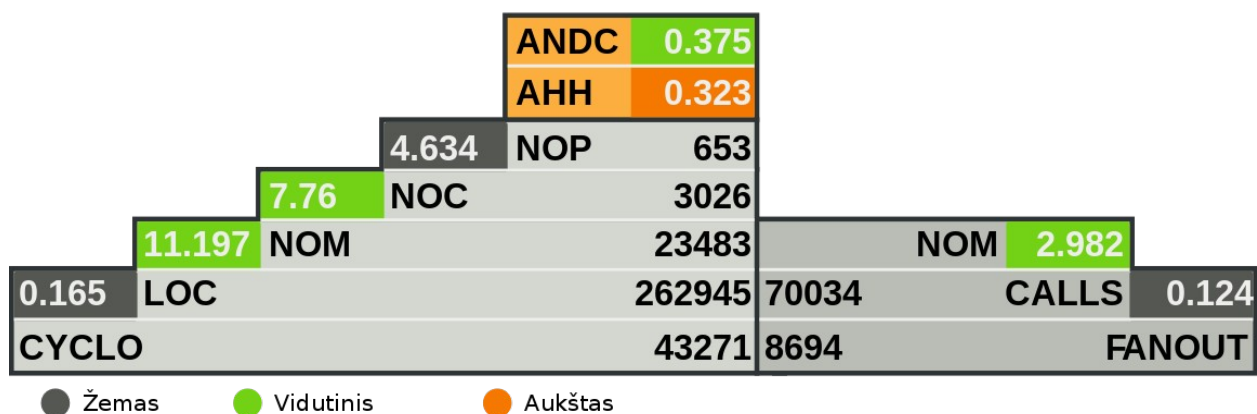
Apžvalgos piramidėje matomi trijų kategorijų rezultatai.

Dydžio ir sudėtingumo kategorijos metrikose matoma, jog LOC/NOM gaunamas gana didelis skaičius, tad tai reiškia, jog atsižvelgiant į kodo eilučių kiekį, metodų kiekis yra gana mažas. Tada galima daryti išvadą, jog Yii karkaso metodai gali būti gana masyvūs.

Iš jungiamumo kategorijoje pateiktų rezultatų galima spręsti, jog sąryšiai tarp metodų nėra sudėtingi.

Iš paveldėjimo kategorijos metrikų matoma, jog klasių paveldėjamumas yra gana aukštas ir vykdyti paties karkaso testavimą gali būti gana sudėtinga.

4.5.3 Zend 2 karkaso kodo analizės tyrimo rezultatai

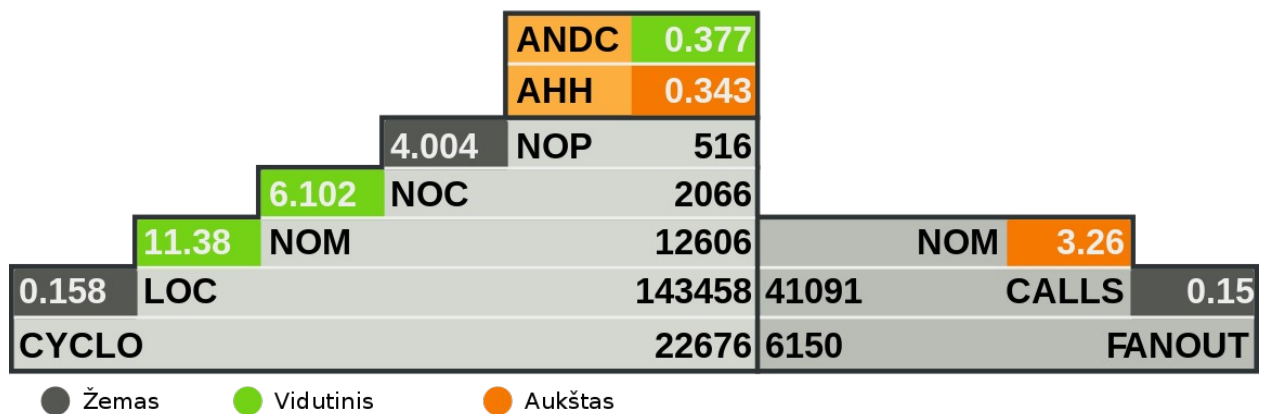


Pav. 21: Zend 2 karkaso tyrimo rezultatai pateikti apžvalgos piramide

Pav. 21 pateikta Zend 2 karkaso apžvalgos piramidė. Peržiūrint, karkaso dydžio ir sudėtingumo kategorijos metrikas matoma, jog ciklomatinis sudėtingumas yra labai mažas. Kitos metrikos taip pat stebina. Karkaso sudėtingumas yra žemas arba vidutiniškas.

Lyginant su Yii karkasu paveldėjimo kategoriją matoma, jog čia klasių paveldėjimo gylis išlieka aukštas, tačiau čia paveldimų klasių vidurkis yra vidutinis.

4.5.4 Symfony2 karkaso kodo analizės tyrimo rezultatai



Pav. 22: Symfony2 karkaso tyrimo rezultatai pateikti apžvalgos piramide

Pav. 22 pateikta Symfony2 karkaso apžvalgos piramidė. Rezultatai labai panašūs į Zend 2 karkaso apžvalgos piramidės rezultatus, tačiau čia CALLS/NOM yra gana aukštas iš ko galima spręsti, jog kreipinių metoduose į kitas funkcijas yra daroma gana daug, neišskirstant į kitus metodus, tad funkcijos gali būti sudėtingesnės ir sunkiau suprantamos.

5 IŠVADOS

- Šiuo metu PHP karkasų kiekis yra milžiniškas, tačiau nėra viešai publikuojamų eksperimentinių tyrimų palyginančių juos tarpusavyje, todėl darbas yra aktualus įmonėms ir programuotojams besirenkantiems PHP karkasus naujiems, bei esamiems projektams.
- Siekiant ištirti pagrindines karkasų charakteristikas buvo pasirinkti Yii, Zend 2 ir Symfony2 karkasai kurie yra dažniausiai naudojami kuriant PHP projektus. Eksperimentų metu buvo numatytas jų darbo imlumas, našumas, funkcionalumas ir atlikta kodo analizė.
- Tyrimo tikslais buvo sukurta QR kodų generavimo ir vartotojų lankomumo sekimo programinė įranga su kiekvienu iš pasirinktų karkasų.
- Atliktas darbo imlumo tyrimas(žr. 4.2 poskyrį). Tirtu atveju pastebėta, kad Yii karkasas yra lengviausiai suprantamas ir su juo sukurti mažą ar vidutinišką sistemą galima greičiausiai. Tačiau dėl šio karkaso modelių struktūros palaikymo metu sistemos sudėtingumas gali labai išaugti. Tad galvojant apie didelę sistemą, kurios gyvavimo laikas bus apie 10 metų, būtų patartina rinktis kitą karkasą.
- Atlikus karkasų našumo tyrimą(žr. 4.3 poskyrį) pastebėta jog Zend 2 karkasas maždaug du kartus lenkia Symfony2 karkasą savo greičiu, tačiau Yii karkasas buvo maždaug 3 kartus greitesnis už Zend 2. Darbo rezultatai buvo panašūs ir atlikus tyrimą įjungus APC. Tad galima teigti, jog Yii karkasas yra greičiausias karkasas lyginant su Symfony2, bei Zend 2 karkasais.
- Atlikus funkcionalumo tyrimą(žr. 4.4 poskyrį) pastebėta, jog kiekvienas iš karkasų turi pakankamai daug įvairių įrankių, tačiau didžioji dalis funkcionalumo priklauso nuo karkaso bendruomenės, kuri viešina sukurtus įskiepius, modulius ar paketus. Kol kas didžiausią bendruomenę turi Symfony2 karkasas, kuo paaiškinamas ir platus funkcionalumas tad galima būtų teigti, kad šis karkasas savo funkcionalumu lenkia likusius.
- Atlikus karkasų kodo analizę(žr. 4.5 poskyrį), pastebėta, jog Zend 2 ir Symfony2 karkasai yra gana panašūs, jų sudėtingumas yra žemas, tačiau Yii karkaso sudėtingumas yra aukščiausias, tad galima teigti, jog šios programinės įrangos kūrėjams, nustojus vykdyti jo atnaujinimo darbus, karkasą būtų sudėtinga palaikyti.
- Atlikus darbe aprašytus tyrimus galime rekomenduoti Yii karkasą dėl jo greičio, bei darbo našumo, mažesniems ar vidutiniškiems projektams, tačiau esant reikalui kurti didesnę projektą, reiktų rinktis Zend 2 ar Symfony2. Reikia nepamiršti, jog Symfony2 karkasas yra suintegruotas su trečiųjų šalių įrankiais ir juos išmokus, programinės įrangos kūrimo greitis būtų žymiai spartesnis, nei kuriant su Zend 2. Taip pat svarbu yra tai, kad Symfony2 bendruomenė yra didžiausia, tad šis karkasas turi ir daugiausiai viešai platinamų nemokamų paketų.
- Sukurta programinė įranga yra teikiama atviro kodo bendruomenei, ją galima pasiekti GitHub svetainėje[9], Taigi tiek tyrinėtojai, tiek entuziastai gali peržiūrėti tyrimus, bei naudoti ir praplėsti šią tyrinėjamąją programinę įrangą papildomiems tyrimams.

6 LITERATŪRA

[1] Remi Woler „Living Apart Together: Decoupling Code and Framework,,. 2012, [Žiūrėta 2013 03 05], prieiga internete :

<<http://phpmaster.com/living-apart-together-decoupling-code-and-framework/>>

[2] Timothy Boronczyk „6 Things to Consider when Choosing a Framework,,. 2013, [Žiūrėta 2013 03 05], prieiga internete: <<http://phpmaster.com/6-things-to-consider-when-choosing-a-framework/>>

[3] David Connelly „Concerns about Zend Framework 2, Symfony2 and other major PHP frameworks,,. 2012, [Žiūrėta 2013 03 05], prieiga internete:

<<http://davidjconnelly.wordpress.com/2012/11/28/concerns-about-zend-framework-2-symfony2-and-other-major-php-frameworks/>>

[4] Wojciech Sznepka „Is Symfony2 a MVC framework?,,. 2012, [Žiūrėta 2013 03 05], prieiga internete: <<http://blog.sznepka.pl/is-symfony2-a-mvc-framework/>>

[5] Kai Chan „Choose the right PHP framework,,. 2013, [Žiūrėta 2013 03 05], prieiga internete:

<<http://www.netmagazine.com/features/choose-right-php-framework>>

[6] Umair Abid „Comparison of Yii Framework and Zend Framework 2,,. 2013, [Žiūrėta 2013 04 20], prieiga internete:

<<http://www.techknowlogists.com/comparison-of-yii-framework-and-zend-framework-2/>>

[7] Steve Floyd „Five of the Best PHP Frameworks,,. 2013, [Žiūrėta 2013 04 20], prieiga internete:

<<http://www.axzm.com/best-php-frameworks>>

[8] Manuel Pichler „Overview Pyramid,,. 2013, [Žiūrėta 2013 04 20], prieiga internete:

<<http://pdepend.org/documentation/handbook/reports/overview-pyramid.html>>

[9] Mantas Vaitkūnas „Free Business app,,. 2012, [Žiūrėta 2013 05 20], prieiga internete:

<<https://github.com/mantas88/freeBusinessapp>>