

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
Informacijos sistemų katedra

**Renaldas Ridlauskas**

**TURINIO VALDYMO SISTEMOS MODELIS**

Magistro darbas

**Vadovas**  
**dr. doc. R. Butleris**

**KAUNAS, 2006**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
Informacijos sistemų katedra

**TURINIO VALDYMO SISTEMOS MODELIS**

Magistro darbas

Vadovas  
dr. doc. R. Butleris  
2006-01-10

Recenzentas:  
doc. dr. E. Karčiauskas  
2006-01-10

Atliko:  
IFM 0/4 gr. stud.  
R. Ridlauskas  
2006-01-10

**KAUNAS, 2006**

# Turinys

1.	Įvadas	7
2.	Turinio valdymo sistemų funkcionalumo ir technologinių sprendimų analizė	8
2.1	Analizės tikslas	8
2.2	Tyrimo sritis, objektas ir problema	8
2.3	Turinio valdymo sistemos vartotojų tipai, tikslai ir problemos	8
2.4	Turinio valdymo sistemos funkcijų analizė	9
2.5	Turinio valdymo sistemų analizė Lietuvos mastu	13
2.5.1	InterTVS turinio valdymo sistema	13
2.5.2	SAGE internetas turinio valdymo sistema	14
2.5.3	EasyWeb turinio valdymo sistemos funkcionalumas ir savybės	14
2.6	Turinio valdymo sistemų analizė pasaulio mastu	17
2.7	Turinio valdymo sistemų palyginimas	19
2.8	Galimų TVS įgyvendinimo priemonių variantų analizė	20
2.9	„Zope“ serverio karkaso koncepcija ir architektūra	22
2.9.1	Objektų publikavimas	23
2.9.2	Fundamentalūs „Zope“ komponentai	23
2.10	Darbo tikslas ir siejami privalumai	25
2.11	Funkciniai reikalavimai ir apribojimai	25
2.11.1	Reikalavimai prieigų administravimui ir nustatymams	25
2.11.2	Tinklapių ir svetainės maketavimo galimybės	26
2.11.3	Dizaino projektavimas	27
2.11.4	Papildomų elementų sukūrimas	27
2.11.5	Sistemos navigacijos projektavimas	28
2.11.6	Paieškos projektavimas	28
2.11.7	Pagrindinės turinio valdymo sistemos savybės	28
2.11.8	Papildomų modulių įdiegimas	28
2.11.9	Reikalavimai serveriui	29
2.11.10	Reikalavimai Web serveriui	29
2.12	Nefunkciniai reikalavimai ir apribojimai	29
2.13	Analizės išvados	29
3.	Turinio valdymo sistemos modelis	31
3.1	Turinio valdymo sistemos modelio kūrimo pagrindimas ir esmės išdėstymas	31
3.2	Turinio valdymo sistemos modelio architektūra	31
3.2.1	Bendras kuriamos sistemos aprašymas	31
3.2.2	Sistemos kontekstas	32
3.2.3	Panaudojimo atvejų vaizdas	32
3.2.4	Turinio valdymo sistemos vartotojų paslaugos	35
3.2.5	Paskirstymo vaizdas	36
3.3	Sistemos elgsenos modelis	37
3.4	Duomenų bazės „ZODB“ architektūra	38
4.	Turinio valdymo sistemos įdiegimo ir testavimo pavyzdys	41
5.	Eksperimentinis sistemos vertinimas	46

5.1	Eksperimentinio TVS diegimo aprašymas	46
5.2	Sukurto turinio valdymo sistemos modelio funkcionalumo eksperimentinis vertinimas	46
6.	Sistemos naudojimo instrukcija	49
6.1.1	Šablono struktūros pritaikymas HTML dokumentui	49
6.1.2	Pagrindinio šablono struktūra (main_template)	49
6.1.3	Išorinių CSS duomenų failų naudojimas	51
6.1.4	Dinaminių elementų įterpimas į šabloną	52
6.1.5	Objektų pavadinimų ir metaduomenų įterpimas	53
6.2	Dinaminiai elementai	53
6.2.1	Dinamiškai kintanti meniu juosta	53
6.2.2	Lokali navigacija	54
6.2.3	Naršymo kelias	54
6.2.4	Kalbų pasirinkimas	54
6.2.5	Paieškos laukas	55
6.2.6	Svetainės planas	55
6.2.7	Tarnybinis prisijungimas	56
6.2.8	Administravimo įrankių blokas	56
6.2.9	Pranešimai vartotojui	56
6.2.10	„Corp” kalendorius	56
7.	Išvados	57
8.	Literatūra	58
9.	Terminų ir santrumpų žodynas	60
	Priedas 1	61
	Priedas 2	64

## Lentelių sąrašas

1 lentelė. TVS sąrašas ir savybės pasaulio mastu.....	17
2 lentelė. TVS palyginimas.....	20
3 lentelė. Eksperimentis TVS vertinimas.....	47
4 lentelė. Koduotės poveikis skirtingų simbolių atvaizdavimui.....	65

## Paveikslėlių sąrašas

1 pav. Turinio valdymo sistemos redagavimo dalies galimybės.....	12
2 pav. Turinio valdymo sistemos pateikimo dalies galimybės.....	12
3 pav. Fundamentalūs „Zope“ komponentai .....	24
4 pav. Turinio valdymo sistemos panaudojimo atvejų vaizdas.....	32
5 pav. Naudojimosi sistema diagrama.....	33
6 pav. Kontekstinė diagrama .....	34
7 pav. Neregistruoto vartotojo naudojimosi sistema diagrama .....	35
8 pav. Sistemos administravimas .....	35
9 pav. Tinklalo dizaino tvarkymas .....	36
10 pav. Paskirstymo diagrama.....	36
11 pav. Sistemos elgsenos diagrama .....	37
12 pav. ZODB architektūra .....	38
13 pav. Objektų būsenų diagrama ir elgsena.....	39
14 pav. „Zope“ serverio administravimo langas .....	41
15 pav. Svetainės su TVS kūrimas .....	43
16 pav. Svetainės vaizdas .....	44
17 pav. Svetainės su TVS bylos .....	44
18 pav. Svetainė sukurta su TVS.....	45
19 pav. Trasų diagrama .....	61
20 pav. Naudojimosi sistema trasų diagrama .....	62
21 pav. Naudojimosi sistema trasų diagrama.....	62
22 pav. Sistemos administravimo trasų diagrama .....	63
23 pav. Karkaso įrankių diagrama.....	64
24 pav. Karkaso objektų diagrama .....	65

## **SUMMARY**

### **THE MODEL OF CONTENT MANAGEMENT SYSTEM**

The content management system is a tool which is essential to load and administrate information that is in the net site.

The main CMS purpose is money and time saving which is committed for the site care. To look after the site and freshen it could an employee not having specific news.

Content management system is analyzing Lithuania's and the world's measure in this work. Their basal internal are reviewing and specifying requests for content management system's model. So, there is analyzing two platforms: "PHP" and "Zope". Considering to every platform's advantages and disadvantages there is arbitrary "Zope" platform.

Considering to the analysis part which specific create the content management system model. So, the content management system accomplished testing model and gives auditorial example. CMS functional is evaluating in the experimental part.

Finally, there is no general purpose control system; there is only theoretical project which shows content management system's place.

# 1. Įvadas

Šiuo metu, kai daugelis įmonių, kuriančių svetaines turi susikūrusios turinio valdymo sistemas (TVS), programavimo darbus pakeičia sistemos įdiegimo ir adaptavimo kliento poreikiams darbai[9].

TVS pagrindinė savybė – lengvas svetainės struktūros bei turinio administravimas.

Šiuolaikinėje turinio valdymo sistemoje privalo būti šios galimybės[13]:

- neriboto gylio struktūros kūrimas, valdymas;
- turinio įvedimas nenaudojant HTML žinių;
- neribotas svetainės administratorių, su skirtingomis galimybėmis skaičius;
- detali tinklapio lankomumo statistika;
- dokumentų valdymas (WorkFlow - pateikti, publikuoti, atšaukti, panaikinti).

Kiekviena TVS turi daugiau ar mažiau papildomų modulių, kurių pagalba interneto svetainėje galima įdiegti įvairių papildomų galimybių: elektroninę parduotuvę, diskusijų grupes, apklausas, reklamines antraštes, įvairių struktūruotų duomenų lenteles, sąsają su kitomis informacinėmis sistemomis (vidiniu kompanijos tinklu, CRM)[7] ir t.t..

Pagrindinė TVS nauda – pinigų ir laiko, skirto svetainės priežiūrai, taupymas. Svetainę prižiūrėti galės neturintis specifinių žinių darbuotojas. TVS – daugelyje projektų patikrintas produktas, todėl įdiegiamas labai greitai, o klaidų tikimybė ypač maža.. Iškilus naujiems poreikiams, TVS yra lengvai plečiama. Turinio valdymo sistema - tai įrankis būtinas norint įkelti ir administruoti informaciją pateikiamą interneto svetainėje[10].

Darbo tikslas – išanalizuoti įvairias turinio valdymo sistemas, kūrimo platformos variantus ir realizavimo priemonės, bei specifiuoti reikalavimus turinio valdymo sistemos modeliui. Aprašyti turinio valdymo sistemos modelio projektą ir įvertinti realizuotos turinio valdymo sistemos funkcionalumą atsižvelgiant į reikalavimus specifiukuotus analizės dalyje.

Šiame darbe išanalizuojamos turinio valdymo sistemos tiek Lietuvos, tiek pasaulio mastu. Apžvelgiamos jų bazinės savybės ir specifiukuojami reikalavimai turinio valdymo sistemos modeliui. Taip pats išanalizuojamos dvi platformos: „PHP“ ir „Zope“. Atsižvelgus į kiekvienos platformos privalumus ir trūkumus pasirenkama „Zope“ platforma.

Atsižvelgus į analizės dalyje specifiukuotus reikalavimus sukurtas turinio valdymo sistemos modelis ir jis realizuotas. Taip pat atliktas turinio valdymo sistemos testavimo modelis ir pateiktas kontrolinis pavyzdys. Eksperimentinėje dalyje įvertinamas TVS funkcionalumas.

## **2. Turinio valdymo sistemų funkcionalumo ir technologinių sprendimų analizė**

### **2.1 Analizės tikslas**

Pagrindinis šio darbo tikslas yra suprojektuoti ir realizuoti turinio valdymo sistemą, kuri pasižymėtų geromis plėtimo savybėmis, išsiskirtų funkcionalumu ir būtų konkurencinga savo kaina.

Norint suprojektuoti ir sukurti būtent tokią turinio valdymo sistemą visų pirma išsiaiškinsime, kokie yra vartotojų tipai ir poreikiai, kurie naudosis šia turinio valdymo sistema. Taip pat apžvelgsime panašius produktus tiek Lietuvos, tiek pasaulio mastu. Įvertinę privalumus ir trūkumus nustatysime, kokio sudėtingumo turinio valdymo sistema turėtų būti realizuota iškeltiems uždaviniams tenkinti.

Kita tyrimų pusė būtų ta, kokia platforma bus pasirinkta turinio valdymo sistemos realizavimui. Apžvelgsime dvi platformas: "Zope" ir "PHP" ir palyginsime kiekvienos platformos galimybes.

### **2.2 Tyrimo sritis, objektas ir problema**

Darbo tyrimo sritis: turinio valdymo sistemos.

Tyrimo objektas: turinio valdymo sistemos Lietuvos ir pasaulio mastu ir naujos TVS realizavimo būdai.

Problema: nustatyta, kad yra daugybė faktorių, kurie erzina vartotoją, dėl to jis išeina iš interneto svetainės nesusipažinęs su pristatomomis paslaugomis bei produktais. Todėl siekiant pritraukti vartotojus, interneto svetainė turi būti tvarkinga, ji turi greitai užsikrauti, tekstai joje turi būti lengvai įskaitomi, tekstų stiliai ir spalvos turi būti suderinti, informacija tinklapyje turi būti nuolat atnaujinama, vartotojas turi turėti galimybę susisiekti su įmonės atstovais, naudotis prekių katalogais, užsisakyti paslaugas internetu, naudotis forumais ir t.t. Visas šias problemas nesunku išspręsti pasinaudojus turinio valdymo sistema. Ši sistema išsiskiria savo patikimumu, efektyvumu, lankstumu, naudojamų technologijų naujove, bei modulių įvairove. TVS pagalba net darbuotojai, neturintys programavimo žinių, galės lengvai ir greitai valdyti informaciją tinklapyje.

### **2.3 Turinio valdymo sistemos vartotojų tipai, tikslai ir problemos**

Suprojektuota ir realizuota turinio valdymo sistema galės naudotis visi tinklapyje vartotojai, į kurių bus įdiegta TVS. Informacijos atnaujinimo ir dizaino darbus galės atlikti tik tokias teises turintys vartotojai. Vartotojai skirstomi į:

- Administratorius (atsakingas už TVS įdiegimą, puslapio ir kitų vartotojų sukūrimą,



informacijos atnaujinimą ir keitimą, papildomų modulių įdiegimą);

- Dizaineris (atsakingas už svetainės dizainą, šablonų ir CSS failo redagavimą);
- Galutinis vartotojas (puslapio, į kurį įdiegta TVS, lankytojas).

Vartotojų tikslai:

- TVS būtų lengvai įdiegiama ir prižiūrima;
- Paprastas informacijos atnaujinimas;
- Geros plečiamumo galimybės;
- Konkurencinga kaina;
- TVS patikima, funkcionali ir lanksti.

Vartotojų problemos:

- Dizaineriui, norint pakeisti standartinę puslapio išvaizdą ir dinaminių elementų išsidėstymo vietą, bus reikalingos „Python“ programavimo kalbos žinios šablonų redagavimui.

## **2.4 Turinio valdymo sistemos funkcijų analizė**

### **Šiuolaikinės interneto svetainės kūrimo procesas**

Interneto svetainės kūrimo procesas yra paprastas ir aiškus – panašus į daugelį kitų kūrybinių-gamybinių procesų. Svarbu, kad kūrybos procese dalyvautų žmogus, išmanantis interneto subtilybes. Teisingas svetainės kūrimo procesas turėtų būti toks:

- Poreikių, tikslų identifikavimas, koncepcijos, preliminarus turinio ir struktūros sukūrimas;
- Dizaino darbai;
- HTML paruošimas;
- Programavimo darbai;
- Turinio įkėlimas.

Šiais laikais, kai didelė dalis įmonių, kuriančių svetaines turi susikūrusios turinio valdymo sistemas (TVS), programavimo darbus pakeičia sistemos įdiegimo ir adaptavimo kliento poreikiams darbai[9].

## **Kas yra turinio valdymo sistema?**

TVS pagrindinė savybė – lengvas svetainės struktūros bei turinio administravimas.

Šiuolaikinėje turinio valdymo sistemoje privalo būti šios galimybės:

- neriboto gylio struktūros kūrimas, valdymas;
- turinio įvedimas nenaudojant HTML žinių;
- neribotas svetainės administratorių, su skirtingomis galimybėmis skaičius;
- detali tinklapio lankomumo statistika.
- Kiekviena TVS turi daugiau ar mažiau papildomų modulių, kurių pagalba interneto svetainėje galima įdiegti įvairių papildomų galimybių: elektroninę parduotuvę, diskusijų grupes, apklausas, reklamines antraštes, įvairių struktūruotų duomenų lenteles, sąsają su kitomis informacinėmis sistemomis (vidiniu kompanijos tinklu, CRM) ir t.t..

## **Kokią naudą gauname įsigydami TVS?**

Pagrindinė TVS nauda – pinigų ir laiko skirto svetainės priežiūrai taupymas. Jūsų svetainę prižiūrėti galės neturintis specifinių žinių darbuotojas. TVS – daugelyje projektų patikrintas produktas, todėl įdiegiama labai greitai, o klaidų tikimybė ypač maža. TVS suteikia galimybę gauti labai detalią statistiką apie Jūsų tinklapio lankomumą, analizuoti kiekvieno lankytojo kelią – iš kur jis atėjo, kokiuose svetainės puslapiuose apsilankė, kiek laiko juose praleido[10]. Iškilus naujiems poreikiams, TVS yra lengvai plečiamos. Turinio valdymo sistema – tai įrankis būtinas norint įkelti ir administruoti informaciją pateikiamą interneto svetainėje.

## **Trumpas TVS galimybių aprašymas**

- Puslapių redagavimas nereikalauja jokių programavimo žinių;
- Galimybė naudoti skirtingus dizaino elementus kuriant vidinius puslapius;
- Integruotas WYSIWYG redaktorius su galimybe kopijuoti tekstus tiesiai iš MS Office;
- Galimybė sistemą prijungti į jau sukurtus tinklapius;
- Dinamiškai kintanti navigacijos struktūra ir svetainės žemėlapis;
- Skirtingų vartotojų lygių ir teisių prisijungimai;
- Integruotas dokumentų valdymas (WorkFlow – pateikti, publikuoti, atšaukti, panaikinti);

- Galimybė nustatyti dokumentų publikavimo datą ir laiką;
- Norint atnaujinti tinklapio dizainą, nereikia perprogramuoti interneto svetainės, ar naujai įkelti informacijos;
- Neribotas vidinių puslapių skaičius ir meniu gylis;
- Papildomų modulių prijungimas;

## **Pagrindinės TVS funkcijos**

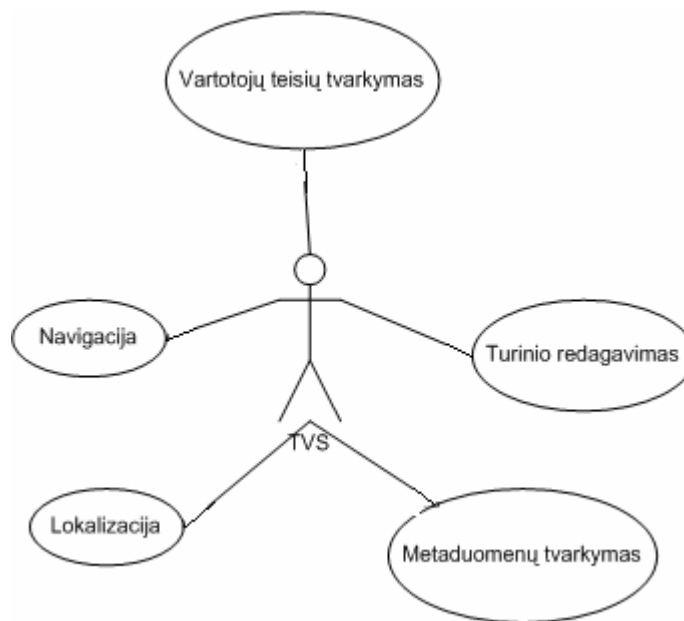
Lankytojui patraukli svetainė su nesudėtingu jos atnaujinimu turi turėti šias savybes[8]:

- Originalų dizainą;
- Galimybę pačiam administruoti svetainę;
- Patogų ir visiems suprantamą informacijos išdėstymą;
- Paiešką svetainėje;
- Galimybė modifikuoti svetainės struktūrą pagal poreikius;
- Papildomų svetainės komponentų įdiegimą į bendrą sistemą.

Plačios interneto svetainės komponentų praplėtimo galimybės leis bet kada įdiegti naujus modulius, nekeičiant svetainės pagrindo. Tai ypač naudinga, nes visa informacija, sukaupta per ilgą laiko tarpą, išliks nepakitusi, kai tuo tarpu visa svetainės struktūra bei išvaizda galės būti neatpažįstamai pakeista ir atnaujinta. Įvairūs papildomi moduliai praplės galimybes ir taip gausų bazinės sistemos variantą .

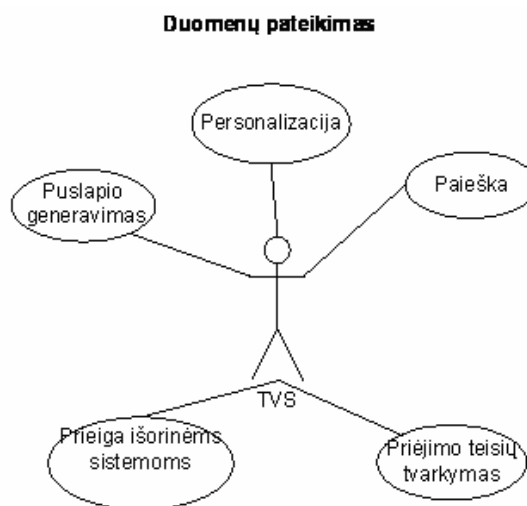
Turinio valdymo sistemų paslaugas galima išskirti į dvi aiškias kategorijas: duomenų apdorojimą ir duomenų pateikimą [3].

**Duomenų apdorojimo** paslaugos skirtos vidiniams sistemos vartotojui, kurio jurisdikcijoje yra duomenų įvedimo bei apdorojimo darbai. Saugaus prisijungimo užtikrinimas – vartotojo autorizacija ir teisių tikrinimas realizuojama per vartotojo grupes: kiekvienas vartotojas priklauso tam tikrai grupei, kuri apibrėžia skirtingas naudojimosi sistema teises. Autoriai pateikia medžiagą ir ją redaguoja, o administratoriai gali redaguoti sistemos kodą, taigi keisti išvaizdą ir galimybes. Informacijos pateikimas TVS sistemai apibūdina darbų seką: pateikimas, redagavimas, patvirtinimas. Versijas skirtingoms kalboms vadiname lokalizacija – pritaikymu atskiros lokalizacijos vartotojams.



**1 pav. Turinio valdymo sistemos redagavimo dalies galimybės**

- **Duomenų pateikimas** remiasi puslapio generavimu. Naudojant stilių šablonus (stylesheets) tinklalapio dizaino elementus galima labai lengvai pakeisti. Papildomos funkcijos pagerina TVS patogumą, greitaveiką bei pasiekiamumą. Daug duomenų apimančiuose tinklalapiuose, informacijos pasiekimo patogumui praverčia paieška. Norint padidinti sistemos veikimo greitį naudojamas kešavimas. Informaciją taip pat galima perduoti kitiems portalams, pvz., naujienų santrauką XML formatu[12].



**2 pav. Turinio valdymo sistemos pateikimo dalies galimybės**

## 2.5 Turinio valdymo sistemų analizė Lietuvos mastu

Kuriant turinio valdymo sistemą išanalizuojami panašių sistemų realizavimo principai ir būdai. Apžvelgsime esamas turinio valdymo sistemas Lietuvos mastu. Išanalizuojame jų veikimo principą, pagrindines savybes, kurias akcentuoja kūrėjai, plečiamumą bei funkcionalumą.

### 2.5.1 InterTVS turinio valdymo sistema

**InterTVS** turinio valdymo sistema sukurta „Zope” platformos pagrindu. Pagrindinės bazinės InterTVS savybės yra[6]:

- Puslapių redagavimas nereikalauja jokių programavimo žinių;
- Galimybė naudoti skirtingus dizaino elementus kuriant vidinius puslapius;
- Integruotas WYSIWYG redaktorius su galimybe kopijuoti tekstus tiesiai iš MS Office;
- Galimybė sistemą prijungti į jau sukurtus tinklapius;
- Integruota objektinė duomenų bazė;
- Dinamiškai kintanti navigacijos struktūra ir svetainės žemėlapis;
- Skirtingų vartotojų lygių ir teisių prisijungimai;
- Integruotas dokumentų valdymas (WorkFlow - pateikti, publikuoti, atšaukti, panaikinti);
- Galimybė nustatyti dokumentų publikavimo datą ir laiką;
- Norint atnaujinti tinklapio dizainą, nereikia perprogramuoti interneto svetainės, ar naujai įkelti informacijos;
- Neribotas vidinių puslapių skaičius ir meniu gylis;
- Rezervinio kopijavimo, viso projekto įkėlimo ir iškėlimo (import/export) galimybės;
- Papildomų Python modulių prijungimas;
- Pilnai palaiko Unicode šriftų koduotes;
- Puslapiai vienodai atrodo nepriklausomai nuo naudojamos naršyklės.

Bazinės IterTVS turinio valdymo sistemos funkcionalumas pritaikytas vizitinės kortelės/bukleto tipo svetainėms ir suteikia jų savininkams galimybę keliomis kalbomis susikurti nuo kelių iki keliolikos puslapių interneto svetainę. Tekstai gali turėti hipertekstines nuorodas, būti iliustruojami paveikslais, informacijai struktūrizuoti gali būti naudojamos lentelės. Savininkams suteikiami metaduomenų redagavimo įrankiai (jų prireikia optimizuojant svetainės geresnėms pozicijoms paieškų mašinos).

## 2.5.2 SAGE internetas turinio valdymo sistema

**SAGE internetas** turinio valdymo sistema leidžia savarankiškai tvarkyti tinklalapio tekstinę ir grafinę informaciją įvairiomis kalbomis, keisti dizainą, vykdyti elektroninę prekybą Internetu. Turinio atnaujinimas neapima jokių aplinkybių. Sistema yra ypač paprasta ir patogi naudoti. Nereikia jokių specialių žinių. Įdiegus sistemą, apmokymai, konsultacijos ir vartotojo instrukcija pateikiami nemokamai. SAGE internetas turinio valdymo sistemos pagrindinės galimybės yra[7]:

- Pritaikoma įvairiam dizainui;
- Informacijos struktūros (menu) valdymas;
- Teksto ir grafikos redagavimas;
- Produktų (prekių) katalogo administravimas (katalogo, užsakymų, sąskaitų, mokėjimų, nuolaidų valdymas; kainynų importavimas; produktų palyginimas, komplektacija, rodyti/nerodyti funkcija; pirkėjo krepšelis ir kt.);
- Skaičiuoklės (lizingo, kainų ir kt.);
- Grafikos ir bylų saugykla;
- Atsarginės tinklalapio turinio kopijos;
- Neribotos kalbų versijos;
- Naujienų administravimas (aktualios informacijos publikavimas, redagavimas, archyvavimas; "bėganti" naujienų eilutė; prenumerata, prenumeratorių sąrašo valdymas);
- Dizaino šablonų valdymas;
- Reklaminių skydelių (angl. banner) valdymas;
- Komentarų, apklausų, balsavimų galimybės;
- Forumas;
- Uždaros vartotojų zonos;
- Klientų valdymo sistemos (angl. CRM) integracija;
- Administratoriai ir jų teisių nustatymas.

## 2.5.3 EasyWeb turinio valdymo sistemos funkcionalumas ir savybės

### Vartotojo sąsaja (interfeisas)

Vartotojo sąsaja (interfeisas) yra optimizuota asmeniui, mokančiam dirbti standartinėje Windows terpėje. Sisteminės komandos, formų laukai, meniu ir kiti elementai išdėstyti taip, kad būtų lengvai suprantami ir pasiekiami. Veiksmų paaiškinimai ir vartotojo pagalba pateikiami papildomai[17].

### **Aktyvus turinio redagavimas (WYSIWYG – What you see is what you get)**

Turinio redagavimas pritaikytas vartotojui, neturinčiam specialių HTML kalbos žinių. Mokančiam naudotis tekstų redaktoriumi vartotojui (pvz., Microsoft Word), nebus sunku administruoti svetainės turinį, nes vartotojo sąsaja labai panaši.

Pagrindinės savybės:

- Šrifto dydžio, spalvos, tipo keitimas, teksto lygiavimas
- Tekstų papildymas grafiniais objektais (nuotraukos/iliustracijos sumažinamos iki reikiamo dydžio), nuorodomis (galimybė įterpti reikiamą nuorodą, nurodant adresą ir pavadinimą), failais (galimybė pridėti dokumentą atskirame faile), lentelėmis (keičiamas lentelės elementų aukštis ir plotis).
- Informacinių objektų kilnojimas į kitą puslapio vietą bei jų modeliavimas reikiama tvarka.
- Papildomų informacijos sričių formavimas bei administravimas. Sistemoje yra numatyta galimybė formuoti papildomas aktualios informacijos sritis. Tai dienos ar savaitės citatų, reklaminių intarpų, paveikslėlių automatizuotas išvedimas bei administravimas.

### **Objektų galiojimas ir skelbimas**

Ši sistemos funkcija, leidžianti autoriui nurodyti viešai skelbiamo dokumento galiojimo terminą, yra automatizuota. Galimi šie veiksmai pasibaigus dokumento galiojimo laikui:

ištrynimasis iš sistemos, deaktivavimas (paliekama sistemoje, bet nerodoma viešai), perkėlimas į archyvą. Artėjant dokumento galiojimo termino pabaigai sistema el. Paštu išsiunčia redaktoriui atitinkamą pranešimą – priminimą.

Taip pat vartotojas gali nurodyti ar pakeisti dokumento galiojimo pradžios datą bei papildomus veiksmus (pavyzdžiui, pranešimų apie išleistą dokumentą išsiuntimą naujiųjų prenumeratoriams).

### **Nuotraukų biblioteka**

Sistemoje kaupiami grafinių objektų: nuotraukų, schemų, paveikslėlių duomenų bazė. Vėliau ši grafinė medžiaga naudojama tekstams iliustruoti.

### **Dokumentų archyvas**

Sistemoje kaupiami suvestų dokumentų bazė. Atliekama detali dokumento paieška pagal autorių, pavadinimą, datą ir pan.

### **Puslapio maketavimas (layout)**

Vizualus šablonų bei informacinių blokų maketavimas: padėties bei dydžio keitimas, naujo bloko įterpimas.

### **Daugiakalbiškumas**

Galimybė svetainę kurti keliomis kalbomis, neribojant jų kiekio. Kiekvienos kalbos svetainę galima administruoti atskirai.

### **Administravimo veiksmų kaupimas**

Sistemoje kaupiamas kiekvienas administratoriaus veiksmas, todėl galima operatyviai atrasti bei panaikinti klaidų priežastis. Administratorius, užsiprenumeravęs gali gauti svarbius sistemos pranešimus (pvz., saugumo) el. Paštu.

### **Teisių sistema**

Sistemoje numatyta galimybė suformuoti bei priskirti tam tikras teises konkrečiai vartotojų grupei ar asmeniui. Teisių sistema koreguojama leidžiant ar uždraudžiant atlikti tam tikrus veiksmus (peržiūrėti, sukurti, ištrinti, koreguoti, perkelti objektą).

Registruoti vartotojai gali būti priskirti šioms standartinėms vartotojų grupėms:

- Lankytojas;
- Klientas;




- Partneris;
- Darbuotojas;
- Administratorius;
- Super-administratorius.

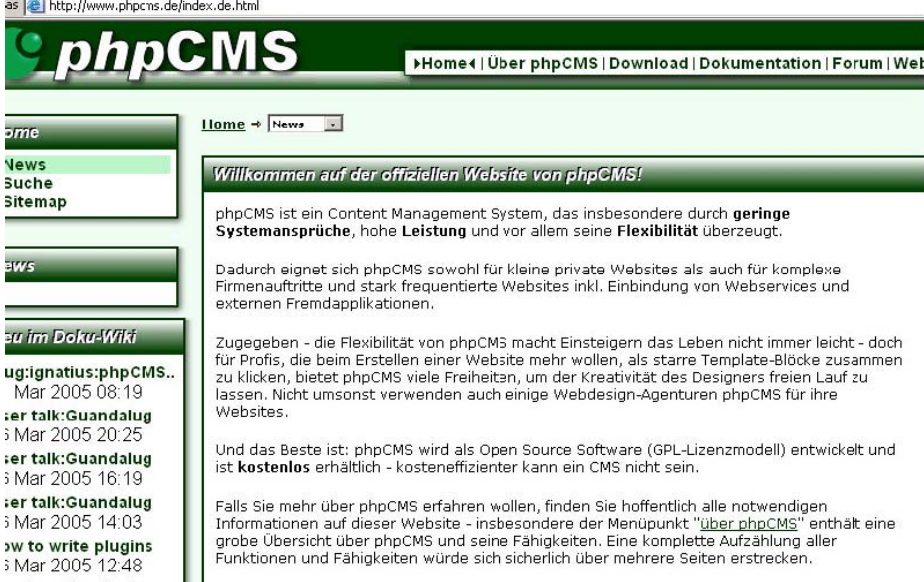
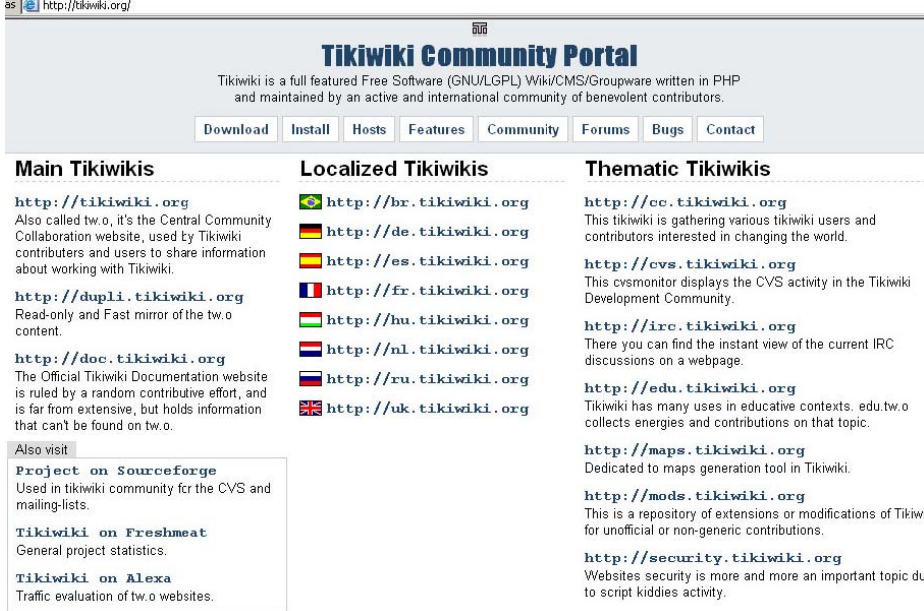
Teisių sistema leidžia kiekvienam įmonės darbuotojui tvarkyti svetainės turinį pagal jam suteiktas teises bei einamų pareigų statusą.

## 2.6 Turinio valdymo sistemų analizė pasaulio mastu

1 lentelėje pateikiamas turinio valdymo sistemų sąrašas pasaulio mastu ir jų pagrindinės savybės[2].

1 lentelė. TVS sąrašas ir savybės pasaulio mastu

Eil. Nr.	TVS	Savybės
1.	<p><a href="http://www.cmsimple.dk/">http://www.cmsimple.dk/</a></p> 	<p>Pati paprasčiausia ir minimaliausia. Visa sukurta svetainė saugoma viename html faile. Turi interaktyvų-vizualų html kodo redaktorių – WYSIWYG (PHP).</p>

Eil. Nr.	TVS	Savybės
2.		<p>Akcentuoja duomenų ir jų vaizdavimo atskyrimą. Turi originalių sprendimų svetainės navigacijos tvarkyme. Pagrindinė dokumentacija tik angliškai. TVS kūrėjai – vokiečiai. PHP, savas šablonų mechanizmas.</p>
3.		<p>Turi plačias vartotojų teisių nustatymo galimybes: ne tik moduliui, bet ir to moduliui konkrečiai informacijai. Yra netgi žaidimų modulis. Vartotojų sistema suderinama su LDAP. Labai smulkmeniška, reikia prižiūrėti daug nustatymu: PHP+MySQL+SMARTY.</p>

Eil. Nr.	TVS	Savybės
4.		<p>Aukštas sistemos saugumo lygis. Turi interaktyvų-vizualų vaizdo redaktorių. Tinka projektų eigos sekimui. Java lemia nepriklausomumą nuo operacinės sistemos. Nėra vartotojų grupių. Tai programuotojų kompanijos produktas, todėl nėra aktyvios palaikančios bendruomenės. Java+MySQL arba Oracle.</p>
5.		<p>Galingiausia ir sudėtingiausia. Tai ne vien tik TVS. Tai turinio valdymo karkasas (Framework). Sudėtinga priežiūra. Reikia mokėti mažai paplitusią programavimo kalbą Python. Palaiko įvairias duomenų bazines. Turi integruota objektiškai orientuotą duomenų bazę ZODB. Taip pat turi integruota serverį, tačiau galima naudoti ir Apache.</p>

## 2.7 Turinio valdymo sistemų palyginimas

Vertinant įvairias turinio valdymo sistemas buvo apžvelgti produktai tiek Lietuvos, tiek pasaulio mastu. Sudaryti pagrindiniai reikalavimai, kuriuos turi tenkinti šiuolaikinės turinio valdymo sistemos.

Išpildyti reikalavimai pažymėti pliusu, o atskiriems panaudojimo atvejams nenumatyti veiksmai pažymėti minusu. Sistemų palyginimas pateiktas 2 lentelėje.

**2 lentelė. TVS palyginimas**

TVS Savybės	InterTVS	SAGE	EasyWeb	CMSimple	phpCMS
Daugiakalbiškumas	+	+	+	-	-
HTML šablonai	+	+	+	+	+
Paieška	+	+	+	-	-
Teksto redagavimo redaktorius	+	+	+	+	+
Neribotas vidinių puslapių skaičius ir menu gylis	+	+	+	-	+
Puslapio medis	+	-	-	-	-
Navigacija	+	Papildomai	Papildomai	-	+
Papildomų modulių pridėjimas	+	+	+	+	+

## 2.8 Galimų TVS įgyvendinimo priemonių variantų analizė

Pasaulyje IT rinkoje egzistuoja nemažai turinio valdymo sistemos kūrimo būdų ir priemonių. Pasirinkti vieną tinkamiausią yra gana sudėtingas uždavinys. Nagrinėjant kiekvieną iš jų atkreipiamas dėmesys į jos prieinamumą, naudojimosi ypatumus, suteikiamas galimybes, plečiamumą.

Renkantis kūrimo platformą išanalizuotos dvi technologijos: PHP ir Zope+Python. Galutinis pasirinkimas buvo „Zope“ technologijos dėl plataus kūrimo, palaikymo priemonių ir komponentų pasirinkimo, lengvai pritaikomos atvirojo kodo koncepcijos. PHP užleido pirmaujančias pozicijas dėl silpnosios objekcinio programavimo koncepcijos, didesnio primityvumo[15].

„Zope“ platforma buvo pasirinkta, kadangi ji leidžia patogiai kurti nedideles interneto svetaines, ir turi rimtų privalumų, kai reikia dirbti su didesniais projektais. Lyginant su PHP, Zope+Python turi geresnę vidinę logiką ir integruotus įrankius, padedančius ne tik efektyviau dirbti, bet taip pat ir lengviau prižiūrėti, suprasti jau sukurtą kodą.

Keletas svarbiausių Zope+Python platformos privalumų:

1. Formulator – klasės darbu su web formomis. Leidžia lengvai kurti praktiškai bet kokio sudėtingumo formas, atskiria formos duomenų struktūrą ir įvestų duomenų patikrinimo procedūras nuo išvaizdos. Web formų apdorojimas – tai tikriausiai didžiausias vargas web

projektuose, ir Formulator leidžia sumažinti jį iki minimumo.

2. ZPT šablonai (Zope page template) leidžia efektyviai atskirti išvaizdą nuo logikos. Tai labai svarbu, nes leidžia programuotojui lengviau suprasti programos logiką, kai kode nesimaišo išvaizdos elementai, ir keisti išvaizdą, nesugadinant programos logikos. Visa tai pagreitina programavimo procesą ir sumažina klaidų skaičių.
3. Lokalizavimo įrankiai leidžia lengvai kurti daugiakalbes svetaines. Kadangi pranešimų katalogai yra atskirti nuo kodo ir yra standartinio formato, su jais nepriklausomai nuo programuotojų gali dirbti atskiri vertėjai. Vertimų sistema integruota su šablonais, tai leidžia versti tiek statinius, tiek ir dinaminčius pranešimus tiesiai šablone.
4. Python+Zope leidžia kurti programas MVC (Model-View-Controller) principu. Vadovaujantis šiuo principu, visas projektas dalinamas į tris dalis: Model – tai pilnas programos funkcionalumas, be jokių vartotojo interfeiso požymių. Model dalis programuojama grynai Python'u. Nepriklausomybė nuo vartotojo interfeiso leidžia automatiškai testuoti modelį (vartotojo interfeiso elementų automatiškai testuoti beveik neįmanoma). View – tai sistemos dalis, atvaizduojanti duomenis ir bendraujanti su vartotoju. Ji realizuojama Zope šablonų pagalba. O Controller – tai klėjai, kurie perduoda duomenis nuo atvaizdavimo modeliui ir atgal. Tokiu būdu modelis „nieko nežino“ apie vartotojo interfeisą, o atvaizdavimas nieko nežino apie biznio logiką ir duomenų bazes. Todėl pakeitimai vienoje iš tų dalių gali būti atliekami nepriklausomai nuo kitų.  
Zope atstovauja Controller ir View dalis. Zope padaro modelio objektus pasiekiamus per HTTP protokolą, suteikiant jiems išvaizdą ZPT (Zope page template) pagalba. Visa tai, teisingai suprojektavus, leidžia automatiškai testuoti visą biznio logiką ir tuo pat metu lanksčiai taikytis prie užsakovo, vartotojo interfeiso poreikių. Tokia schema reikalauja gero kalbos objektinio modelio, kurio PHP4 neturi.
5. Python turi puikius automatinio testavimo mechanizmus, kurie leidžia parašyti testus bet kokiam ne vartotojo interfeiso kodui. Kai tokiu būdu testuoto kodo yra pakankamai daug, unittest'ai leidžia anksti aptikti klaidas, kurios atsirado po kokio nors programos pakeitimo arba kitos klaidos pataisymo. Nors testų rašymas reikalauja papildomų programuotojų pastangų, jie žymiai padidina kodo kokybę ir palaikomumą.
6. Beveik visos funkcijos, kurias Zope leidžia pasiekti per naršyklę, automatiškai palaiko XML-RPC protokolą, išsaugant visas saugumo prielaidas. Tai leidžia beveik be papildomų pastangų realizuoti turtingą API ir suteikti vartotojams galimybę programiškai pasiekti ir manipuluoti tais pačiais duomenimis, kurie pasiekiami per naršyklę.

Dėl operatorių perkrovimo (operator overloading) trūkumo ir silpno tipizavimo PHP kalboje, XML-RPC – tiek serverio, tiek ir kliento – realizavimas yra labai painus [14].

7. Zope turi gana lankstų publikuojamų objektų prieigos teisių mechanizmą. Galima apsaugoti tiek pačius objektus, tiek ir atskiras jų funkcijas. Prieigos teisės čia yra deklaruojamos, o ne programuojamos – o tai sumažina klaidų tikimybę.

Visos šios savybės būtinos, arba naudingos dideliems projektams. Bazinis PHP neturi šių savybių. Daugelis iš šių priemonių yra sukurtos ir PHP kalbos pagrindu, tačiau jų įsisavinimas nėra paprastesnis negu Zope alternatyvų, o dažnai net sudėtingesnis dėl PHP kalbos dizaino trūkumų, kurie dažnai neleidžia paslėpti nuo vartotojų (šiuo atveju programuotojų) realizavimo savybių.

PHP įrankiai yra nepriklausomi ir reikalauja papildomų integravimo pastangų. Pavyzdžiui yra daug šablonų bibliotekų, bet vienos yra nepatogios, kitos nepalaiko vertimų ir t.t. Galiausiai, projektuojant didelį projektą, vienos ar kitos programavimo kalbos žinojimas neturi didelės įtakos. Įtakos turi objektinio projektavimo patirtis ir darbo su framework'ais patirtis.

## 2.9 „Zope“ serverio karkaso koncepcija ir architektūra

Zope“ (Z Object Publishing Environment) yra daugiaplatforminis interneto taikomųjų programų (web application) serveris, įgalinantis greitai kurti patikimas žiniatinklio programas. „Zope“ yra parašytas „Python“, labai produktyvia objektiškai orientuota skriptų, kalba [1].

„Zope“ skirtas turinio valdymo sistemų, portalų, įprastų taikomųjų programų kūrimui. Jis lengvai integruojamas su dauguma RDBMS („Oracle“, „PostgreSQL“, „MySQL“), tokiu būdu suteikdamas funkcionalumą, palyginamą su PHP ar ASP bei dar geresnį. „Zope“ palaiko HTTP, XML-RPC, FTP, WebDAV protokolus.

Pagrindinės savybės:

- Duomenys saugomi objektinėje duomenų bazėje (ZODB), bet galima naudotis ir kitais duomenų šaltiniais (failų sistemos, reliacinių duombazių ir t.t.).
- Sistemą galima keisti per webinį interfeisą (taip vadinamas TTW – Through The Web – vystymas), o taip pat galima pasiekti per FTP arba WebDAV.
- Galima nustatyti priėjimą skirtingiems vartotojams su skirtingais priėjimo lygiais.
- Dinaminiai Web puslapiai: ZPT (Zope Page Templates) arba DTML.

## 2.9.1 Objektų publikavimas

Technologija, kurią „Zope“ išrado realizavimo metu yra objektiškai orientuota. HTTP protokolas suteikia galimybę nusiųsti žinutę objektui ir persiųsti to objekto atsakymą. „Zope“ objektų struktūra yra hierarchinė. Tai reiškia, kad tipiškas „Zope“ puslapis yra sudarytas iš objektų, kurie sudaryti dar iš kitų objektų. Pavyzdžiui adresas „/Marketing/index.html“ gali būti naudojamas objekto-dokumento „index.html“ pasiekimui, kuris yra patalpintas kataloge-objekte „Marketing“. „Zope“ turi galimybę publikuoti sukurtus objektus. Publikavimo principas yra nesudėtingas[1]:

1. Naršyklė nusiunčia užklausimą „Zope“ serveriui. Užklauskas aprašomas URL užklausa tokia forma: protocol://host:port/path?querystring“. Pavyzdžiui:

http://www.zope.org:8080/Resources?batch\_start=100.

2. „Zope“ išskirsto URL užklausą į komponentus „host“, „port“, „path“ ir „query string“ ( http://www.zope.org , 8080 , /Resources and ?batch\_start=100 , respectively).

3. Zope patalpina objektus į objektinę duomenų bazę su atitinkamu komponentu „path“ ( /Resources).

4. „Zope“ sukuria objektą naudojantį „query string“ su tam tikrais parametrais. Užklausa gali modifikuoti objekto elgseną. Tai reiškia, kad objektas gali elgtis kiekvieną kartą skirtingai atsižvelgiant į užklausų reikšmes.

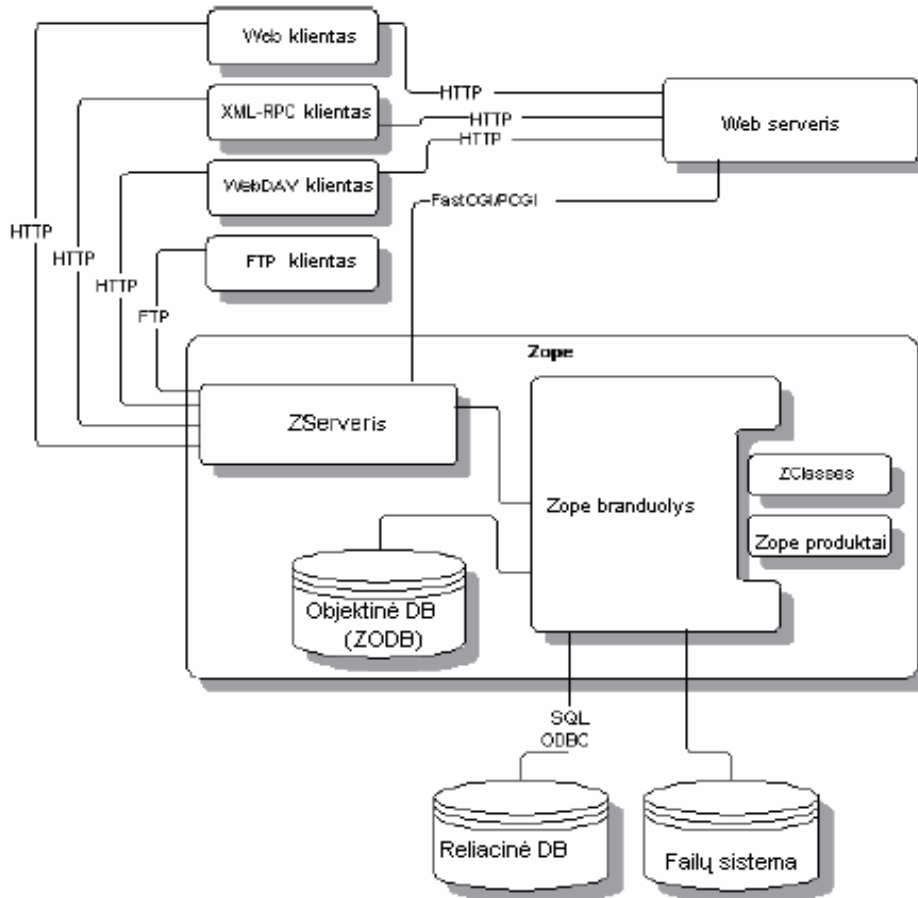
5. Jei objektas gražina kokią nors reikšmę, tai ji yra siunčiamą atgal į naršyklę. Paprastai „Zope“ objektai gražina HTML, duomenų failus ar paveikslėlius.

6. Gražinti duomenys yra interpretuojami naršyklės ir parodomi vartotojui.

Tai nėra nauja idėja. Web serveriai Apache ar Microsoft's IIS viską atlieka lygiai taip pat.

## 2.9.2 Fundamentalūs „Zope“ komponentai

„Zope“ susideda iš keletą skirtingų komponentų, kurie dirbdami kartu padeda kurti Web aplikacijas. „Zope“ fundamentalūs komponentai yra parodyti 3 paveikslėlyje[1]:



**3 pav. Fundamentalūs „Zope” komponentai**

- **Zserveris** — „Zope“ integruotas vidinis Web serveris. Šį serverį galima pasiekti per FTP, WebDAV, ar XML-RPC (a remote procedure call facility).
- **Web serveris** — „Zope“ gali būti naudojamas kartu su *Apache* ar *Microsoft IIS* Web serveriais.
- **Zope branduolys** — tai komponentas, kuris koordinuoja atvaizdavimą, valdymo interfeisą ir objektinę duomenų bazę.
- **Objektinė DB** — integruota „Zope“ objektinė duomenų bazė. Kai naudojamosi „Zope“, paprastai dirbama su objektais, kurie yra patalpinti „Zope“ objektinėje duomenų bazėje
- **Reliacinė DB** — realiacinė duomenų bazė, kurią galima naudoti vietoje ZODB. „Zope“ gali dirbti su *Oracle*, *PostgreSQL*, *Sybase*, *MySQL* ar kitomis reliacinėmis duomenų bazėmis..
- **Failų sistema** — komponentas atsakingas už darbą su dokumentais ir kitais failais patalpintais serverio failų sistemoje.



- *Zclasses* — komponentas, kuris leidžia „Zope“ tinklapio valdytojui pridėti naujus objektus, naudojant Zope Management Interface.
- *Zope produktai* — komponentas, kuris leidžia tinklalapio valdytojui pridėti naujo tipo objektus įdiegiant juos į serverio failų sistemą.

## 2.10 Darbo tikslas ir siekiami privalumai

Pagrindinis darbo tikslas suprojektuoti ir realizuoti tokią turinio valdymo sistemą, kuri būtų patraukli tiek vartotojams, tiek galutiniams naudotojams. Bus siekiama, kad bazinė turinio valdymo sistema savo pagrindinėmis savybėmis ir kaina būtų konkurencinga analogiškomis turinio valdymo sistemoms. Turinio valdymo sistema (TVS) bus orientuota tiek į nepatyrusį vartotoją, tiek ir į profesionalus. Kuriant TVS bus siekiama sukurti patogų ir patrauklų klientui įrankį, kuriuo būtų malonu, paprasta ir greitai dirbti. Pagrindiniai siekiami sistemos privalumai:

- Paprastai ir greitai įsisavinama;
- Nereikia jokių programavimo žinių administruojant informaciją;
- Nedidelių ir vidutinio sudėtingumo tinklapių poreikių tenkinimas;
- Prireikus, lengvai ir greitai plečiama prijungiant papildomus modulius;
- Galimybė naudoti objektinę ar reliacinę duomenų bazę;
- Naudojamas Web serveris “Zope” arba “Apache”;

## 2.11 Funkciniai reikalavimai ir apribojimai

Apžvelgus turinio valdymo sistemas tiek Lietuvos, tiek pasaulio mastu buvo įvertintas jų funkcionalumas, teigiamos ir neigiamos savybės. Pagal tai buvo sudaryti reikalavimai projektuojamai ir realizuojamai turinio valdymo sistemai, kurie detalčiai aprašyti šiame skyrelyje.

### 2.11.1 Reikalavimai prieigų administravimui ir nustatymams

Reikalavimai turinio valdymo sistemos administratoriaus, projektuotojo bei vartotojo sąsajai yra tokie:

- Turinio valdymo sistemos projektuotojo sąsają sudaro teksto redaktorius ir dizaino projektuotojų prieigos;

- Sąsajų skaičius sistemos projektuotojui yra neribojamas;
- Sistemos redaktorius ir sistemos dizaino projektavimo priegios turi būti savarankiškos ir atskirtos nuo kitų sąsajų;
- Turinio valdymo sistemos savininkui turi būti suteikiama teisė pačiam keisti papildyti, pašalinti ar kitaip valdyti TVS palaikomo produkto struktūrą, turinį ir dizainą (informaciją);
- Turinio valdymo sistema leidžia vartotojui/lankytojiui peržiūrėti, kopijuoti visus TVS palaikomo produkto sritis ir puslapius;
- Turi būti numatyta galimybė sistemos ir tinklalapio vartotojų, lankytojų registravimui, registravimo apskaitai, statistikai;
- Turi būti numatytas lankytojams skirtų komentarų, atsiliepimų, vertinimų, prašymų registravimas;

Turinio valdymo sistemos nustatymai:

- Turinio valdymo sistema ir tinklalapio nustatymus gali atlikti sistemos administratorius, sistemos projektuotojai ir savininkas arba jo atstovai;
- TVS nustatymai leidžia keisti:
  - tinklalapio pavadinimą;
  - tinklalapio aprašymą;
  - tinklalapio raktinius žodžius;
  - tinklalapio savininko vardą ir pavadinimą;
  - tinklalapio elektroninį paštą;
  - tinklalapio pagrindinę kalbą;

### **2.11.2 Tinklapių ir svetainės maketavimo galimybės**

Turinio valdymo sistema turi turėti keletą tekstinių puslapių redagavimo galimybių. Galimi teksto redagavimo būdai:

- WYSIWYG, darbas panašus į Word ir Excel redaktorių darbą;
- Tikro HTML kodo redagavimas;
- Struktūrizuotas teksto redagavimas;
- Tiesiog teksto redagavimas.

WYSIWYG redaktoriaus galimybės:

- Rašyti bei redaguoti tekstą.
- Įkelti tekstą iš sukurtų dokumentų MS Word ir MS Excel bei kitų formatų (cut, copy, paste).
- Įkelti nuotraukas ir grafinius vaizdus gif, jpg, png formatuose.
- Sukurti ir redaguoti lenteles.
- Iš anksto paruoštų bylų (failų) įkėlimas (download)
- Atlikti šrifto keitimą:
  - bold, italic, underline;
  - pakeisti šrifto tipą;
  - keisti raidžių aukštį, storį, spalvą.
- Atlikti teksto lygiavimą (į kairę, į dešinę, centras);
- Atlikti dokumentų ir puslapių numeraciją;
- “Atgal” ir “Pirmyn” funkcijos;
- Nuorodų sukūrimas.

### **2.11.3 Dizaino projektavimas**

- Galima sukurti naujų puslapių šablonų dizainą arba pakeisti, modernizuoti turimus šablonus;
- Dizaino modulis leidžia sukurti vaizdinius elementus, juos perkelti, išdėstyti puslapiuose;
- Su dizaino modeliu leidžiama dirbti internetinio tinklalapio savininkui arba jo atstovams. Reikalingos HTML ir Python programavimo kalbų žinojimas.

### **2.11.4 Papildomų elementų sukūrimas**

- Kiekvienas puslapis gali turėti elementų, nepriklausančių nuo puslapio turinio, logotipai, kontaktiniai duomenys, reklaminiai tekstai ir vaizdai;
- Papildomi elementai turi paveldėjimo sąvoką – vidinių puslapių elementai paveldi informaciją iš aukštesnio lygio puslapių, t.y. jiems gali būti priskirta informacija, kurią paveldės gilesnio lygio puslapiai.

## **2.11.5 Sistemos navigacijos projektavimas**

Menu medžio projektavimas ir valdymas:

- Menu medžio automatinis formavimas;
- Menu išdėstymas skirtingose vietose.

## **2.11.6 Paieškos projektavimas**

Turinio valdymo sistema turi turėti integruotą vidinę paieškos sistemą. Paieškos rezultatų detalus ir informatyvus atvaizdavimas.

## **2.11.7 Pagrindinės turinio valdymo sistemos savybės**

- Daugiakalbiškumas;
- Neribotas kuriamų puslapių ir meniu šakų bei elementų skaičius;
- Galima suteikti vidiniam puslapiui alternatyvų, lengvai įsimenamą adresą;
- Projektuojama, redaguojama viename lauke – lauko pildymo principu;
- Yra galimybė automatiškai pakeisti administravimo režimą iš Word į HTML aplinką;
- Integruotas FTP serveris leidžia projektavimo darbus vykdyti asmeniniame kompiuteryje, o vėliau sukurtus puslapius nusiųsti į nuotolinį serverį;
- Paprasta ir logiška sukurtų puslapių navigacija;
- Atskirtos administratoriaus, tekstų redaktoriaus ir dizaino projektuotojo sąsajos;

## **2.11.8 Papildomų modulių įdiegimas**

Ateityje turi būti galimybė įdiegti šiuos modulius:

- Elektroninės komercijos (B2B, B2C);
- Statistikos rinkimo ir apdorojimo;
- Darbo organizavimo;
- Klientų informavimo;
- Užsakymų registravimo;
- Apskaitos;
- Naujienų modulis.

### **2.11.9 Reikalavimai serveriui**

- Serverio operacinė sistema – bet kuri, turinti “Zope“ arba Apache palaikymą;
- Procesorius – Intel Pentium™ arba analogiškas;
- Reikalinga operatyvinė atmintis – 128 MB ir daugiau(rekomenduojama);
- Minimali laisva disko vieta – 100 MB.

### **2.11.10 Reikalavimai Web serveriui**

- Palaikymas Apache 1.3.x arba Zope 7.2.x;
- Duomenų bazė – MySQL 3.23.x arba ZODB, arba PostgreSQL;
- Naršyklė – MS Internet Explorer 5.5 arba vėlesnė versija.

### **2.12 Nefunkciniai reikalavimai ir apribojimai**

- Efektyvus puslapių turinio ir dizaino projektavimas, redagavimas, administravimas;
- Sistema lengvai įsisavinama vartotojų. Puslapiai teisingai formuojami visose naršyklėse;
- Pakankama duomenų apsauga, patogus administravimas;
- Sukurtų internetinių puslapių svetainių struktūra suderinta su žinomiausiomis paieškos sistemomis;
- Išplėtimo galimybės;
- Konkurencinga kaina;
- Nesudėtingas įdiegimas ir palaikymas;
- TVS patikima, funkcionali ir lanksti.

### **2.13 Analizės išvados**

1. Atlikus rinkos analizę, buvo apžvelgtos turinio valdymo sistemos paplitusios Lietuvos ir pasaulio mastu. Išsiaiškinti turinio valdymo sistemos vartotojai, jų poreikiai ir tikslai. Specifikuoti reikalavimai projektuojamam turinio valdymo sistemos modeliui.

2. Renkantis kūrimo platformą, svarstyta keletas alternatyvų: PHP ir Zope+Python technologijos. Galutinis pasirinkimas buvo Zope+Python technologijos dėl plataus kūrimo, palaikymo priemonių ir komponentų pasirinkimo, lengvai pritaikomos atvirojo kodo koncepcijos, nepriklausomumo nuo platformos. PHP užleido pirmaujančias pozicijas dėl silpnesnio objektinio programavimo koncepcijos, didesnio primityvumo. Taip pat Zope+Python platforma buvo pasirinkta dar ir dėl to, kad ji leidžia

patogiai kurti nedideles interneto svetaines, ir turi rimtų privalumų, kai reikia dirbti su didesniais projektais. Lyginant su PHP, Zope+Python turi geresnę vidinę logiką ir integruotus įrankius, padedančius ne tik efektyviau dirbti, bet taip pat ir lengviau prižiūrėti, suprasti jau sukurtą kodą.

3. Nuspręsta suprojektuoti ir realizuoti turinio valdymo sistemą pasižyminčią savybėmis, specifikuotomis analizės dalyje. TVS bus plačių galimybių, paprastai įdiegiama, lengvai administruojama.

### **3. Turinio valdymo sistemos modelis**

Šiame skyriuje aprašomi projektuojamos ir realizuojamos turinio valdymo sistemos reikalavimai, apibrėžiama veiklos sfera. Taip pat aprašomas architektūrinis projektas įvairiais aspektais. Toliau pateikiamos svarbiausios realizacijos metu sudarytos diagramos.

#### **3.1 Turinio valdymo sistemos modelio kūrimo pagrindimas ir esmės išdėstymas**

Projektuojama svetainės turinio valdymo sistema yra skirta mažai ir vidutinio dydžio svetainei, kuri pilnai įgalins valdyti turimą internetinę svetainę, t.y. keisti visą grafinę bei tekstinę informaciją. Šis programinis paketas gali būti įdiegtas tiek į naujai kuriamą svetainę, tiek ir į jau sukurtą anksčiau. Tai universali, paprasta ir patogi turinio valdymo sistema.

Realizuojant ir projektuojant turinio valdymo sistemą buvo panaudoti moduliavimo ir integravimo principai. Branduolyje sukauptos bendros visom sistemom reikalingos savybės ir numatytos lengvos specializuotų modulių integravimo galimybės. Ypatingas dėmesys buvo skiriamas produkto kokybei, darbo su sistema paprastumui ir greitumui. Vertinant sistemą, galima pabrėžti greitą sistemos įdiegimą bei lengvą ir greitą papildomą modulių integravimą. Novatoriškos idėjos įdiegtos kuriant sistemos branduolį leidžia sistemos vartotojui pasiekti didesnę ekonominę naudą ir efektyviau valdyti besikeičiančius duomenų srautus įmonėje, tarp jos filialų ar tarp bendradarbiaujančių įmonių, sukurti klientų informavimo ar užsakymų priėmimo ir apdorojimo sistemas.

#### **3.2 Turinio valdymo sistemos modelio architektūra**

##### **3.2.1 Bendras kuriamos sistemos aprašymas**

Pagrindinės turinio valdymo sistemos sudedamosios dalys yra objektų duomenų bazė, šablonai ir dinaminiai elementai. TVS leidžia dirbti su šiais objektais – dokumentais, katalogais ir iliustracijomis. Dokumentai ir katalogai interneto naršyklėje yra rodomi kaip savarankiški objektai. Iliustracijos yra įterpiamos į kitus dokumentus. Objektus, naudodamasis vartotojo sąsaja, tvarko pats svetainės savininkas.

Kataloguose yra saugomi dokumentai ir iliustracijos. Dokumentų išskirstymas kataloguose pagal temas žymiai palengvina navigaciją interneto svetainėje. Interneto svetainės krovimosi metu, objektas kreipiasi į interneto svetainės dizaino šabloną ir jame nurodytose vietose (Slots) įrašo vartotojo informaciją – objekto turinį.

Šablonas – tai specifinės struktūros HTML dokumentas. Šis dokumentas yra statiškas svetainės atvaizdas, kuriame nurodyta kaip puslapyje bus išdėstyti vartotojo objektai ir dinaminiai elementai, kaip

jie atrodys ir elgsis, kokios bus naudojamos spalvos, šriftai, nuorodos, rėmeliai, mygtukai. Šablone yra nurodytos vietos (Slots), kuriose bus įrašomas objekto turinys[4].

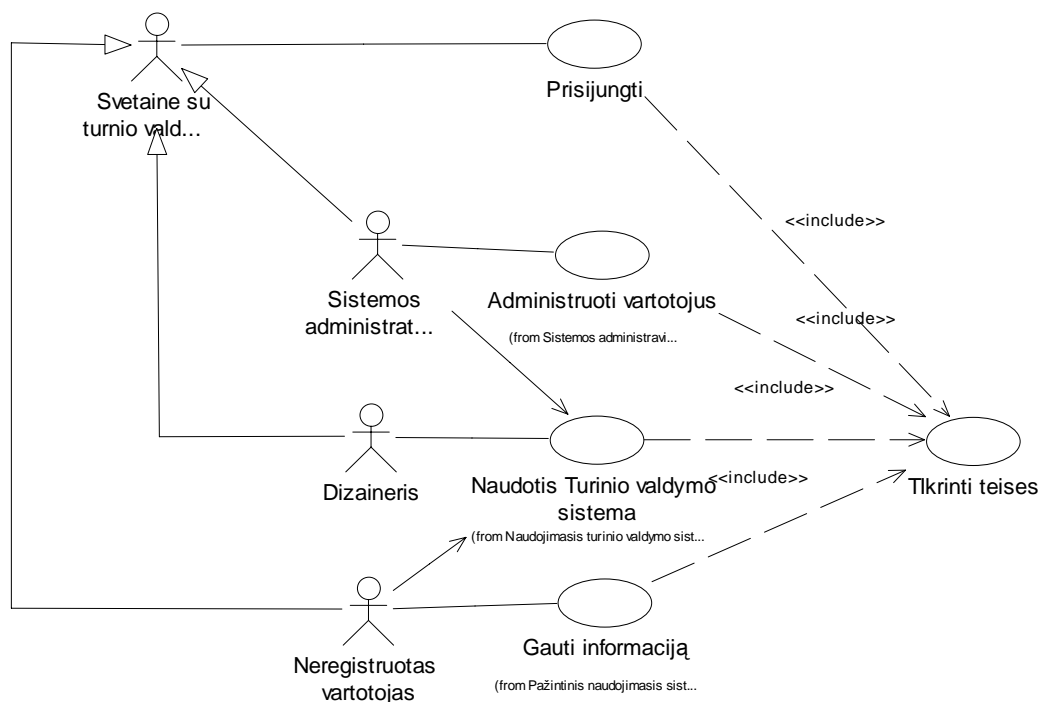
Dinaminiai elementai – paieška, navigaciniai meniu, kalbos pasirinkimo meniu, vartotojų autorizacija, turinio valdymo meniu blokas, tarnybiniai pranešimai. Tai yra atrinktas minimalus Zope funkcijų rinkinys skirtas standartinėms interneto svetainėms.

### 3.2.2 Sistemos kontekstas

Produktas naudojamas vienas. Nėra sistemos dalis. Produktas yra pats kaip sistema, į kurią įeina kiti komponentai. Visi sistemos nustatymai atliekami per interneto sąsają, išskyrus tik patį sistemos įdiegimą.

### 3.2.3 Panaudojimo atvejų vaizdas

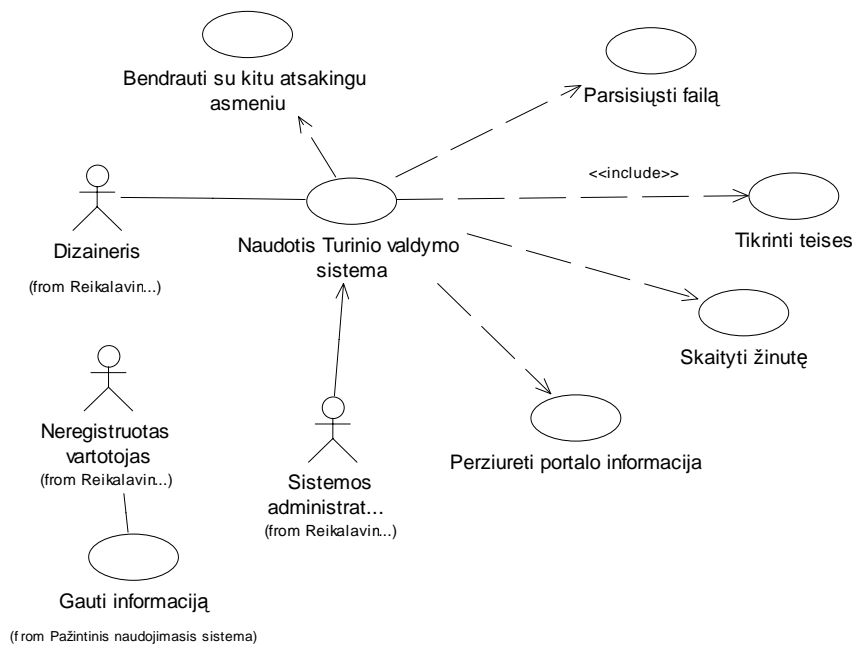
Pasinaudojant atlikta analize buvo išskirtos funkcijos turinio atvaizdavimui, redagavimui ir trynimui. Turinio valdymo sistemos funkcionalumas buvo padalintas trijų lygių vartotojams: sistemos administratorius, dizaineris ir galutinis vartotojas.



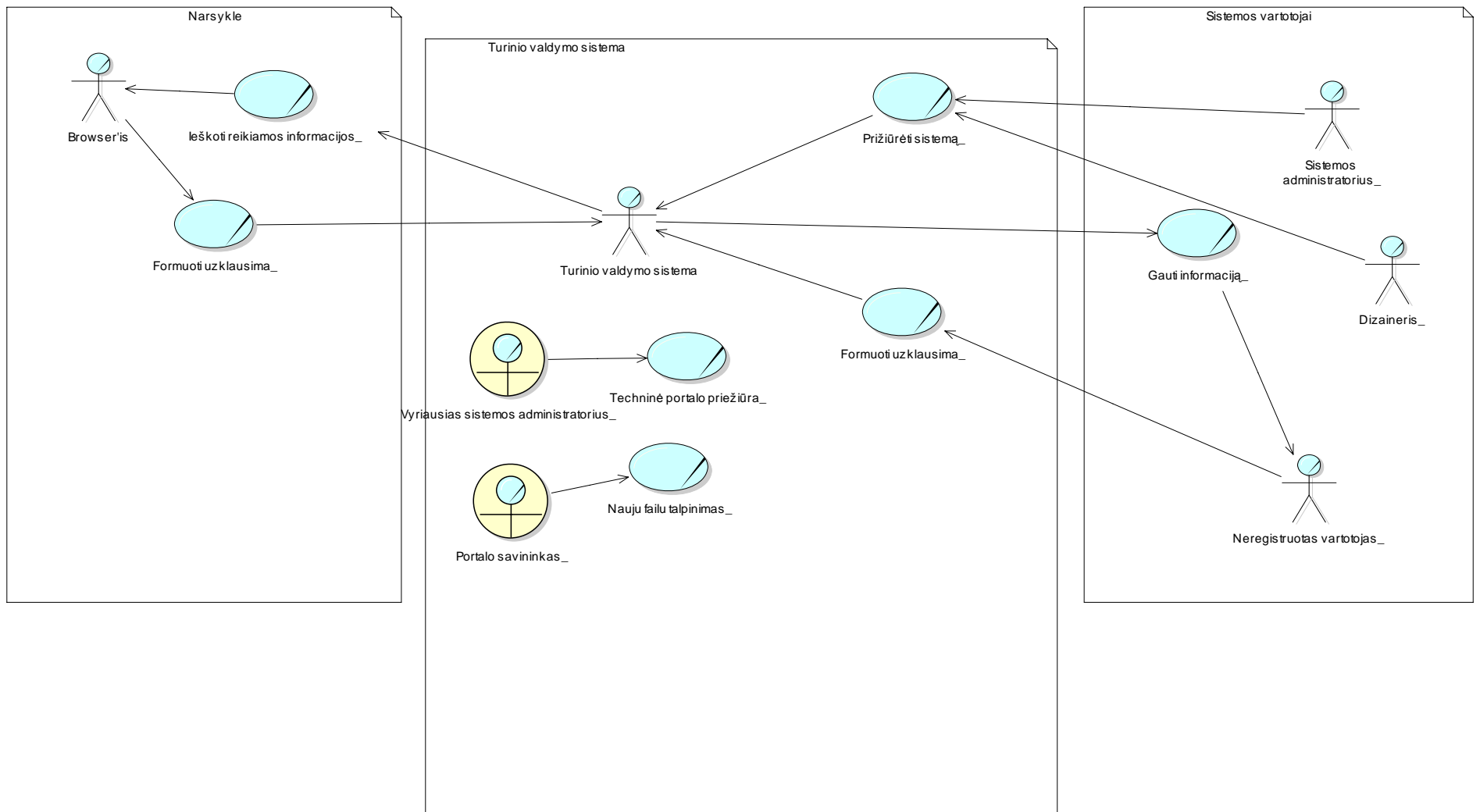
4 pav. Turinio valdymo sistemos panaudojimo atvejų vaizdas



Žemiau esančiame 5 paveikslėlyje pateikiama bendra naudojimosi sistema diagrama.



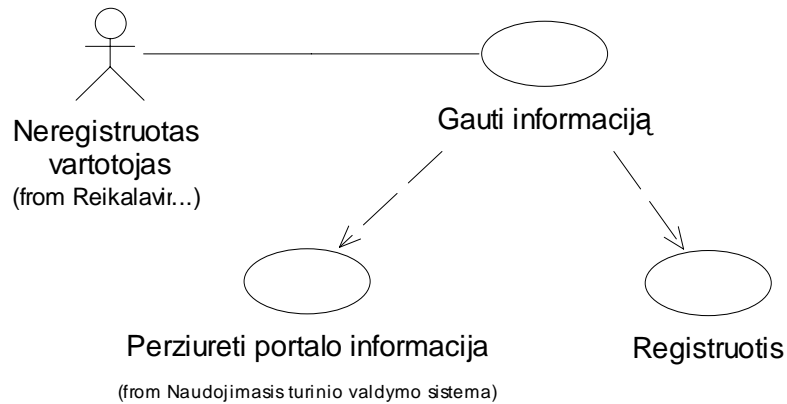
**5 pav. Naudojimosi sistema diagrama**



6 pav. Kontekstinė diagrama

### 3.2.4 Turinio valdymo sistemos vartotojų paslaugos

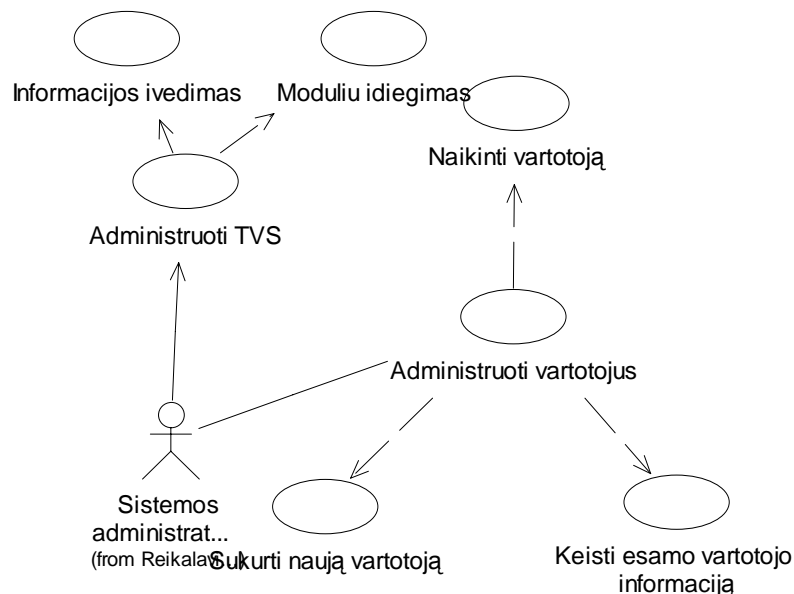
Žemiau pateiktame 7 paveikslėlyje matome galutinio neregistruoto vartotojo naudojimosi sistema diagramą.



7 pav. Neregistruoto vartotojo naudojimosi sistema diagrama

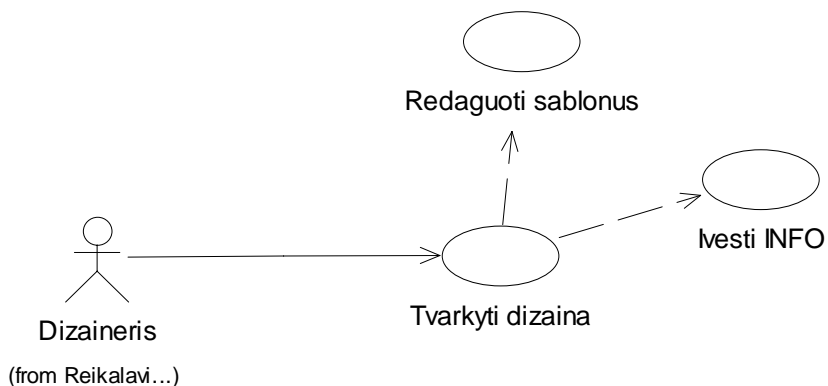
Galutinis vartotojas prisijungęs prie puslapio, kuriame įdiegta TVS, gali skaityti informaciją. Jei administratorius jam sukūręs vartotoją, tuomet jis gali registruotis ir atnaujinti ar keisti internetiniame puslapyje esančią informaciją.

Žemiau esančiuose paveikslėliuose 8 pav. ir 9 pav. pateikiamos sistemos administratoriaus ir dizainerio naudojimosi sistema diagramos.



8 pav. Sistemos administravimas

Sistemos administratorius gali kurti vartotojus, keisti jų statusą. Taip pat jis gali sukurti internetinį puslapį su integruota TVS, įdiegti papildomus modulius, redaguoti informaciją, keisti dizainą.

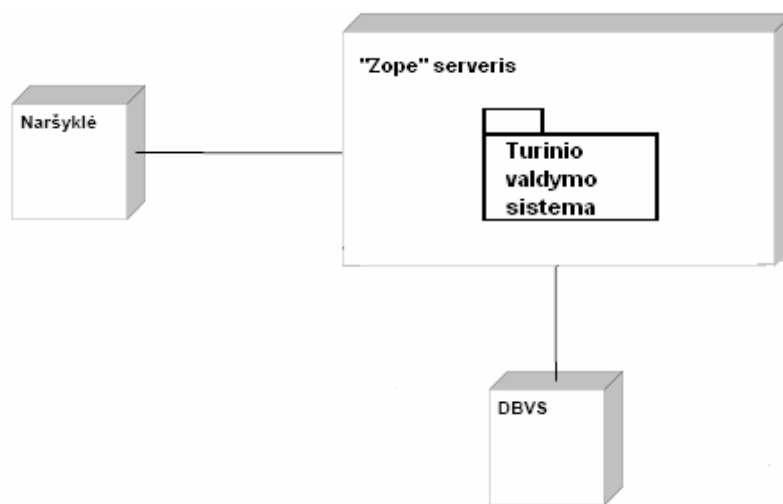


### 9 pav. Tinklalapio dizaino tvarkymas

Dizaineris gali tvarkyti internetinio puslapio dizainą redaguodamas šablonus. Taip pat galimas informacijos redagavimas.

### 3.2.5 Paskirstymo vaizdas

Klasikinį trijų lygių sistemos pasiskirstymo modelį, pritaikome ir šiam inžineriniam sprendimui. Vartotojas duomenis įveda ir rezultatus pamato interneto naršyklėje. Ši bendradarbiauja su programine įranga per programų serverį „Zope“. Duomenys saugomi integruotoje duomenų bazėje ZODB arba kitoje pasirinktoje reliacinėje duomenų bazėje.

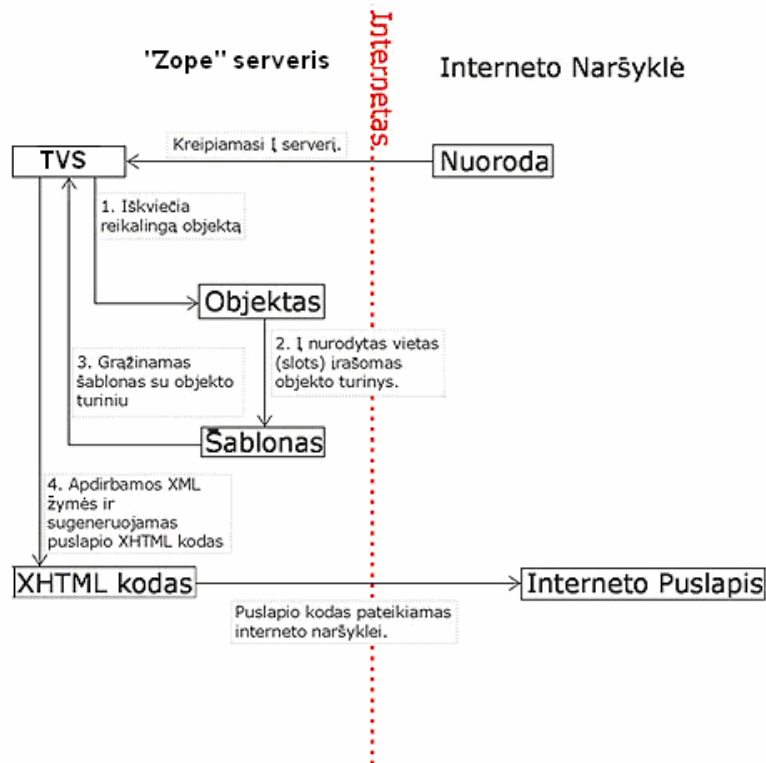


10 pav. Paskirstymo diagrama

### 3.3 Sistemos elgsenos modelis

1. Lankytoji įvedus nuorodą į interneto naršyklės adreso laukelį, naršyklė kreipiasi į „Zope“ serverį. TVS pagal nuorodą, suranda ir iškviečia reikalingą objektą iš objektų duomenų bazės.
2. Objektas į nurodytas šablono vietas (slots) įrašo turinį. Jei objektas yra dokumentas – tuomet turinys yra vartotojo tekstas su iliustracijomis. Jei objektas – katalogas, kuriame nėra index\_html dokumento, tuomet turinys – katalogo pavadinimas, katalogo aprašymas ir nuorodos į kataloge laikomus dokumentus bei kitus katalogus.
3. Užpildytas šablonas yra gražinamas turinio valdymo sistemai.
4. Šablone, naudojantis XML žymėmis (tagais) yra aprašyti dinaminiai (kintantys) elementai. Apdirbus šias žymes, į šabloną yra įrašomas sugeneruotas elementų kodas.
5. Užpildytas šablonas (su objektų turiniu ir dinaminiais elementais) yra konvertuojamas į XHTML 1.0 kodą ir pateikiamas naršyklei, kuri ir užkrauna interneto svetainės puslapį.

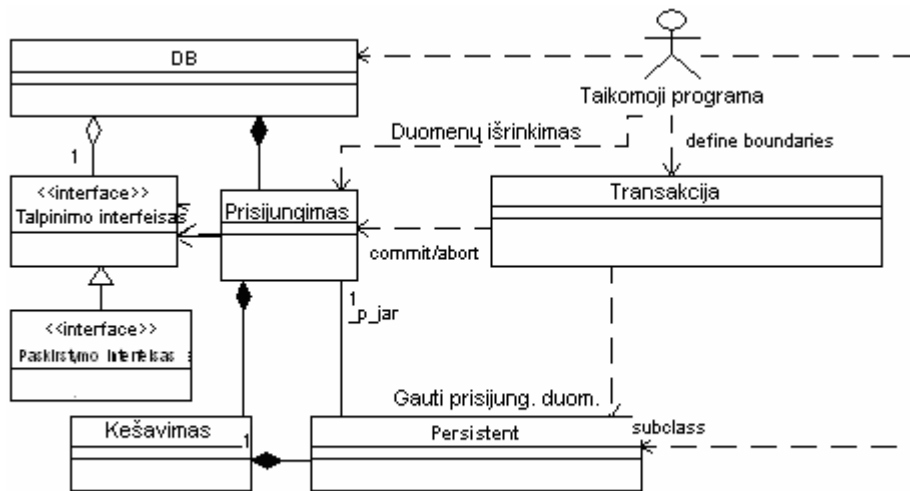
Žemiau pateiktame paveikslėlyje 11 pav. pavaizduota detali sistemos elgsenos diagrama.



11 pav. Sistemos elgsenos diagrama

### 3.4 Duomenų bazės „ZODB“ architektūra

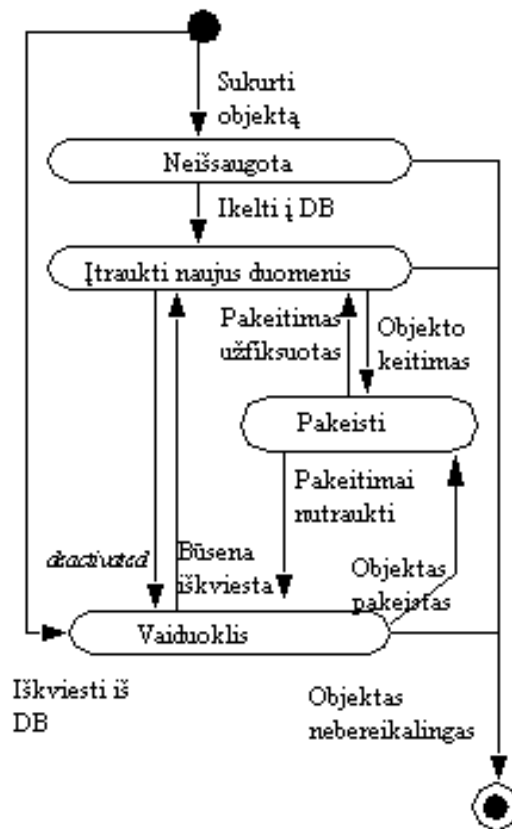
“ZODB” duomenų bazės architektūra pateikta žemiau esančiame paveikslėlyje[5]



12 pav. ZODB architektūra

- **DB** – objektinė duomenų bazė. Ji atsakinga už prisijungimų valdymą ir susiejimą su patalpintais objektais.
- **Talpinimo interfeisas** – klasės, kurios įgyvendina Storage interfeisą, pasirūpina priėjimu konkrečiu žemo lygio talpinimo mechanizmu.
- **Transakcija** – aplikacijos sąveikauja su transakcijų objektais, kad apibūdintų ribas ir nustatytų transakcijų metaduomenis. Kiekvienas transakcijos objektas gali būti naudojamas daugialypėms loginėms transakcijoms.
- **Prisijungimas** – atsakingas už objektų grąžinimą iš objektų saugyklos ir sąveikavimą su transakcijų tvarkytoju.

Žemiau esančiame 13 paveikslėlyje pavaizduota objektų busenų diagrama.



13 pav. Objektų būsenų diagrama ir elgsena

### Neišsaugota

Kai objektas sukuriamas pirmą kartą, jis yra neišsaugotoje būsenoje. Neišsaugotas objektas pereina i būseną "Įtraukti naujus duomenis" (up-to-date), kai patalpinamas i duomenų bazę.

Neišsaugotas objektas neegzistuos kaip ir bet koks kitas „Python“ objektas.

Objektas, kuris buvo išsaugotas duomenų bazėje yra būsenoje „up-to-date“.

### Įtraukti naujus duomenis (up-to-date)

Objektas pakeis būseną tuomet, kai jis bus modifikuotas. Modifikavimo metu objektas yra užregistruojamas perėjimų kontroliavimo sistemos. Taigi objekto būsena gali būti išsaugota kai perėjimas įvykdomas ar sugražinama atgal kaip perėjimas nutraukiamas.

Jei objektas „up-to-date“ būsenoje yra ilgą laiko tarpą ir nebuvo panaudotas, tuomet „cache manager“ gali nuspręsti deaktivuoti objektą. Objektas vis dar bus atmintyje, tačiau pereis i būseną „Vaiduoklis“.

### Pakeisti

Kai objektas pakeistas, jis pereina i būseną „Pakeisti“. Jei perėjimas patvirtintas, tuomet objektas yra nukopijuojamas i duomenų bazę ir objektas pareina i būseną „up-to-date“.

Je perėjimas nutraukiamas, objektas deaktivuojamas ir pereina i būseną

„Vaiduoklis“.

Objektas esantis būsenoje „Vaiduoklis“ vis dar yra atmintyje.

### **Vaiduoklis**

Jei objekto atributas bandomas pakeisti, tai objekto būseną iš DB pereina į būseną „up-to-date“. Taip pat įmanoma nustatyti objekto atributą, kuris yra „vaiduoklio“ būsenoje. Tuomet toks objektas pereina į būseną „Pakeisti“

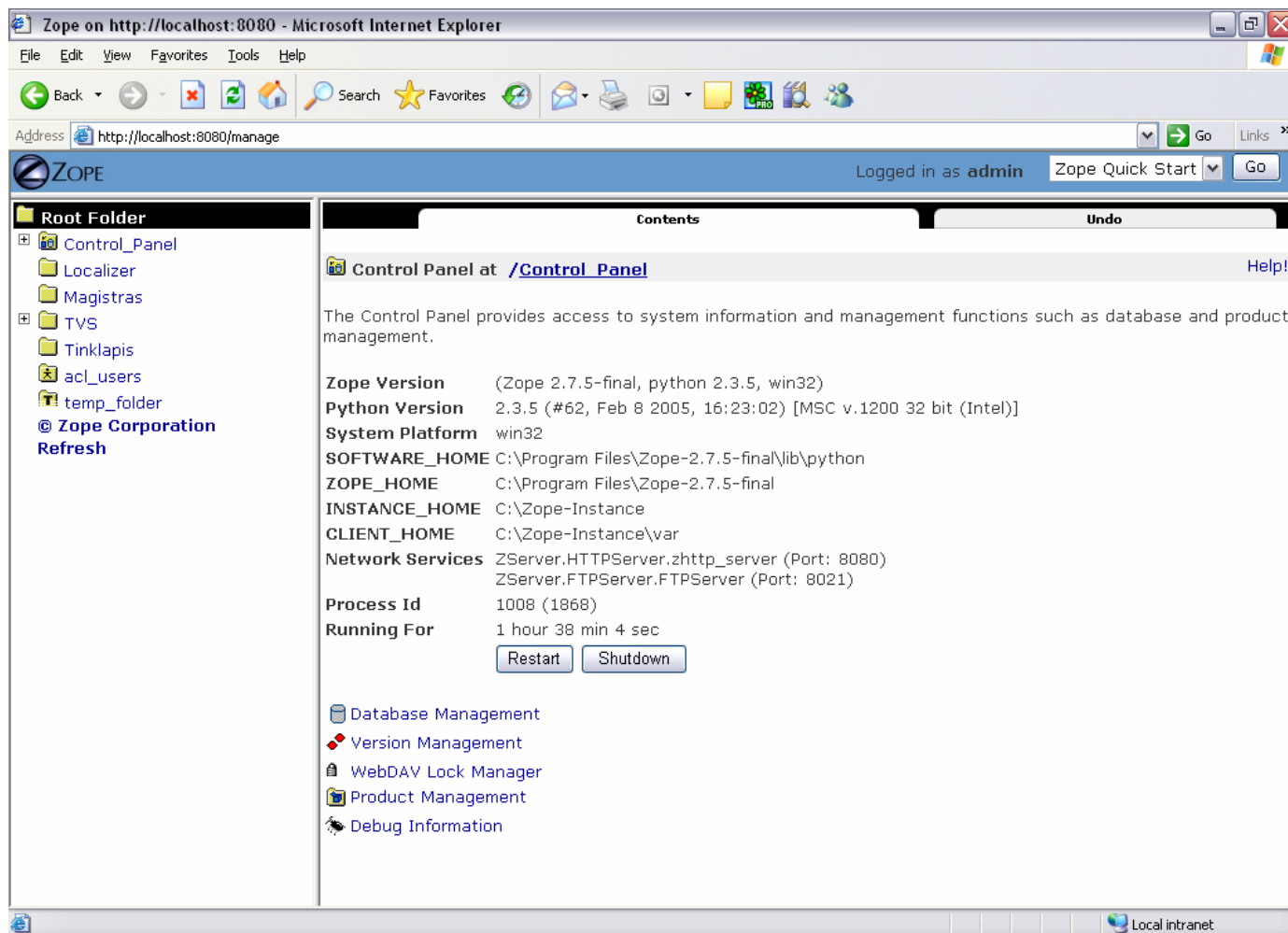
Pagrindinis (root) objektas yra žodynas, kuris laiko visus ilgalaikius objektus. Pvz.  
root['Darbininkai'] = ['Marija', 'Jonas', 'Antanas'].

Trumpai aprašėme magistratūros studijų metu sukurtos turinio valdymo sistemos modelį. Buvo apibrėžtas kuriamo produkto tikslas, bei taikymo sritis. Taip pat buvo detaliam išanalizuotas problemos sprendimas pasaulyje, bei atliktas kruopštus situacijos Lietuvoje įvertinimas. Kuriant sistemą būtina atsirinkti ir priimti teisingus architektūrinius sprendimus, kurie yra trumpai aprašyti šioje dalyje. Šiame skyriuje apžvelgta programinės įrangos techninė – projektinė dalis, padėjo apsispręsti ir priimti pasirinkto sprendimo realizacijos kelią, kurį kaip tik ir pateikėme.



## 4. Turinio valdymo sistemos įdiegimo ir testavimo pavyzdys

Šiame skyrelyje pateiksime suprojektuotos ir realizuotos sistemos testavimo duomenis. Visu pirmaisia į darbo vietą reikia įdiegti „Zope“ serverį. Kadangi testavimui buvo pasirinkta Windows Professional Pro platforma, tai iš gamintojų puslapio <http://www.zope.org> buvo parsisiųstas instaliacinis failas Zope-2.7.5-final-win32.exe. skirtas „Windows“ platformai. Įdiegus serverį prie jo galima prisijungti adresu <http://localhost:8080/manage>. 14 paveikslėlyje pateiktas „Zope“ serverio administravimo langas.



14 pav. „Zope“ serverio administravimo langas

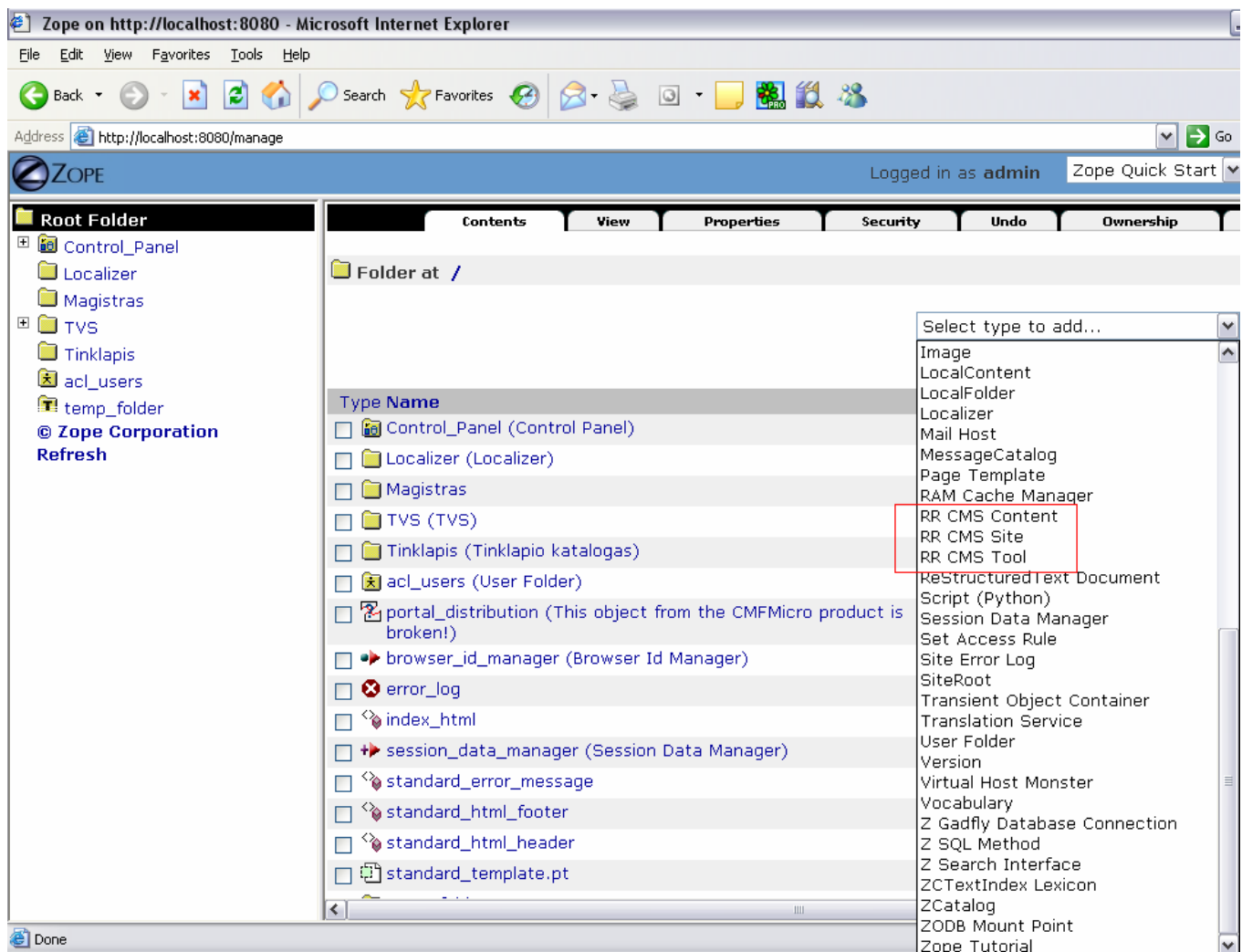
Prieš įdiegiant turinio valdymo sistemą, pirmaisia reikia įdiegti žemiau išvardintus produktus:

- Zope >= 2.6.1 – „Zope“ serveris;
- Zope CMF 1.4.2 – „Zope“ turinio valdymo sistemos karkasas. Tai puiki platforma turinio valdymo sistemai realizuoti;

- Localizer >= 1.0.1 - tai „Zope“ produktas, kuris leidžia sklandžiai plėtoti daugiakalbes Web taikomasias programas;
- TranslationService >= 0.3 – šis „Zope“ produktas leidžia visiškai realizuoti daugiakalbiškumą puslapių šablonuose naudojant standartines i18n žymes;
- Formulator – tai karkaso praplėtimas, kuris palengvina Web formų kūrimą ir įteisinimą;
- Epoz 0.8.2 – tai produktas, kuris leidžia redaguoti „Zope” objektus su „WYSIWYG” redaktoriumi;
- StructuredText i18n patch for zope 2.6.1 - tai produktas, kuris leidžia rašyti struktūrizuota tekstą gimtąja kalba (simboliais, UTF-8 kodavimas);
- CMFFormController – tai produktas, kuris puslapio administratoriui leidžia keisti paketų elgseną be programavimo;
- CMFRR 1.0 – turinio valdymo sistema su bazinėmis savybėmis.

Išvardintų produktų įdiegimas: į direktoriją /Zope/Lib/python/Products/ nukopijuojami visi produktai ir perkraunamas „Zope“ serveris.

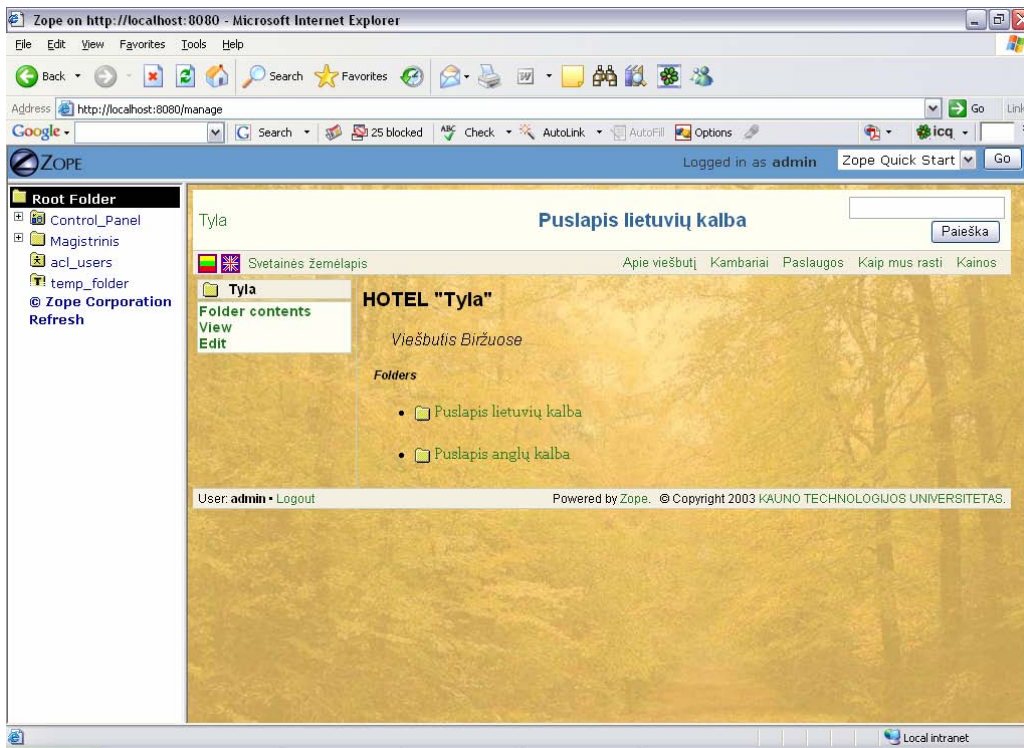
Jei viskas įsidiegė sklandžiai, reikia prisijungti prie „Zope“ serverio (pvz.: <http://localhost:8080/manage>) ir patikrinti – ControlPanel -> ProductManagement sąrašė turėtų atsirasti objektas „CMFRR“. Taip pat sąrašė objektų, kuriuos galima įdėti į „Zope“ objektų duomenų bazę, turi atsirasti objektas „RR CMS Site“. Vaizdas pateiktas 15 paveikslėlyje.



### 15 pav. Svetainės su TVS kūrimas

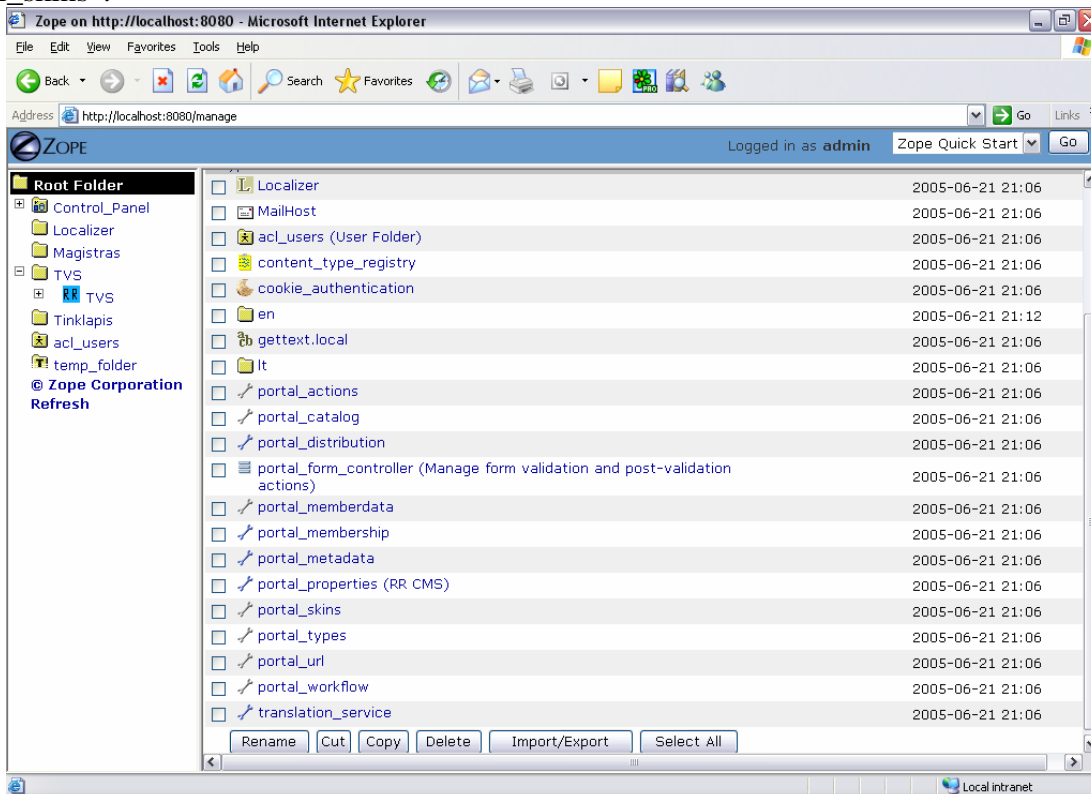
Po to sukuriamas katalogas, į kuri bus patalpinta svetainė valdoma turinio valdymo sistema. Šiuo atveju sukurtas katalogas yra „Magistras“.

Produktų sąrašė pasirenkame ADD RR CMS Site ir toliau suvedame reikalingus duomenis. Paspaudus mygtuką „OK“ sukuriama interneto svetainė su bazinėmis turinio valdymo sistemos savybėmis. Administratorius po įdiegimo jau gali sukurti naujus vartotojus.



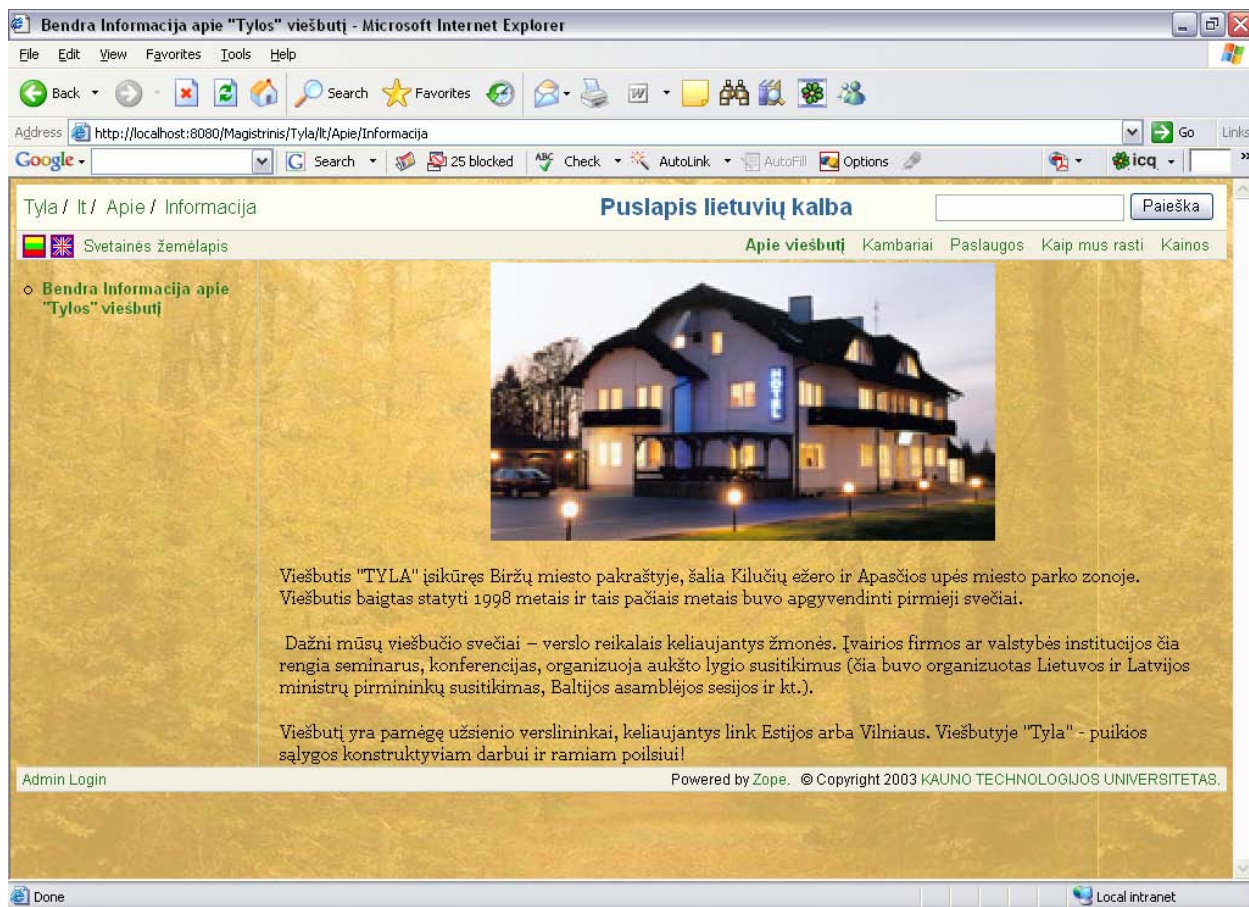
16 pav. Svetainės vaizdas

17 paveikslėlyje pateiktos visos svetainės bylos. Visi pagrindiniai šablonai saugomi kataloge „portal\_skins“.



17 pav. Svetainės su TVS bylos

Galutinis vartotojas prisijungęs naršyklės pagalba prie serverio, kuriame įdiegta TVS, mato pradinį svetainės variantą pateiktą 18 paveikslėlyje. Puslapio dizaineris redaguodamas šablonus ir stilių failą gali keisti svetainės dizainą ir dinaminių elementų išsidėstymo vietą.



18 pav. Svetainė sukurta su TVS

## 5. Eksperimentinis sistemos vertinimas

### 5.1 Eksperimentinio TVS diegimo aprašymas

Prieš įdiegiant turinio valdymo sistemą, pirmiausia reikia įdiegti žemiau išvardintus produktus:

- Zope >= 2.6.1 – „Zope“ serveris;
- Zope CMF 1.4.2 – “Zope” turinio valdymo sistemos karkasas. Tai puiki platforma turinio valdymo sistemai realizuoti;
- Localizer >= 1.0.1 – tai „Zope“ produktas, kuris leidžia sklandžiai plėtoti daugiakalbes Web taikomas programas;
- TranslationService >= 0.3 – šis „Zope“ produktas leidžia visiškai realizuoti daugiakalbiškumą puslapių šablonuose naudojant standartines i18n žymes;
- Formulator – tai karkaso praplėtimas, kuris palengvina Web formų kūrimą ir įteisinimą;
- Epox 0.8.2 – tai produktas, kuris leidžia redaguoti “Zope” objektus su “WYSIWYG” redaktoriumi;
- StructuredText i18n patch for zope 2.6.1 - tai produktas, kuris leidžia rašyti struktūrizuota tekstą gimtąja kalba (simboliais, UTF-8 kodavimas);
- CMFFormController – tai produktas, kuris puslapio administratoriui leidžia keisti paketų elgseną be programavimo;
- CMFRR 1.0 – turinio valdymo sistema su bazinėmis savybėmis.

Išvardintų produktų įdiegimas: į direktoriją /Zope/Lib/python/Products/ nukopijuojami visi produktai ir perkraunamas „Zope“ serveris.

Jei viskas įsidiegė sklandžiai, reikia prisijunkti prie „Zope“ serverio (pvz.: <http://localhost:8080/manage>) ir patikrinti – ControlPanel -> ProductManagement sąrašė turėtų atsirasti objektas „CMFRR“. Taip pat sąrašė objektų, kuriuos galima įdėti į „Zope“ objektų duomenų bazę, turi atsirasti objektas „RR CMS Site“.

### 5.2 Sukurto turinio valdymo sistemos modelio funkcionalumo eksperimentinis vertinimas

Vertinant produktą buvo sudaryti pagrindiniai kriterijai ir taip įvertinamas turinio valdymo sistemos sukūrimo pilnumas.

Tyrimo metu užpildėme sekančią lentelę, kurioje pažymėjome turinio valdymo sistemos atitikimą specifikacijai. Išpildyti specifikuoti reikalavimai pažymėti žodžiu „Realizuota“, o neišpildyti reikalavimai pažymėti minusu.

**3 lentelė. Eksperimentis TVS funkcionalumo įvertinimas**

<b>Eil. Nr.</b>	<b>Funkcionalumas</b>	<b>Įvertinimas</b>	<b>Pastabos</b>
1.	Daugiakalbiškumas	Realizuota	
2.	HTML šablonai	Realizuota	Svetainės gali būti originalios kompozicijos ir dizaino
3.	Paieška	Realizuota	Įgalina lankytoją atlikti paiešką interneto svetainės dokumentuose
4.	Neribotas vidinių puslapių skaičius ir meniu gylis	Realizuota	
5.	Dinamiškai kintantis puslapio medis	Realizuota	
6.	Dinamiškai kintanti navigacija	Realizuota	
7.	Papildomų modulių pridėjimas	Realizuota	Galimybė naudoti papildomus „Zope“ produktus
8.	Darbo organizavimo modulis	Realizuota	Palengvina mažų įmonių darbuotojų tarpusavio bendravimą interneto naršyklės pagalba
9.	ButtonBuilder modulis	Realizuota	
10.	Prisijungimas prie darbo aplinkos	Realizuota	Saugus prisijungimas prie darbo aplinkos
11.	Atsijungimas nuo darbo aplinkos	Realizuota	
12.	B2C modulis	-	
13.	WYSIWYG redaktorius	Realizuota	Teksto redaktorius suteikiantis panašias redagavimo galimybes kaip ir MS Word
15.	Struktūrizuotas teksto redagavimas	Realizuota	
16.	Tiesiog teksto redagavimas	Realizuota	
17.	Išorinių dokumentų įkėlimas	Realizuota	Paveikslėlių, *.doc ar *.xls failų įkėlimas ir publikavimas
18.	Meniu išdėstymas skirtingose vietose	Realizuota	Meniu publikavimas skirtinguose puslapio vietose.
19.	Integruotas FTP serveris	Realizuota	Duomenų perkėlimas FTP protokolu.
20.	Apskaitos modulis	-	
21.	Dokumentų valdymas (WorkFlow – pateikti, publikuoti, atšaukti, panaikinti)	Realizuota	Galimybė dokumentą publikuoti galutiniam

			vartotojui ar publikavimą atšaukti.
22.	Metaduomenų redagavimas	Realizuota	Pritaikyta paieškos mašinoms
23.	CSS failo panaudojimas	Realizuota	Stiliaus failas supaprastina puslapio dizaino tvarkymą
24.	Skirtingų vartotojų teisių nustatymas	Realizuota	
25.	Unicode šifruočių palaikymas	Realizuota	Leidžia tame pačiame dokumente atvaizduoti skirtingose šalyse naudojamus rašmenis

Kalbant apie praplėtimo galimybes – pasirinkta architektūra leidžia nesunkiai padidinti turinio valdymo sistemos galimybes ir paslaugas.

Specifikacijoje numatytos turinio valdymo sistemos savybės buvo išpildytos ir išbandytos. Testavimo modelis plačiau aprašytas skyrelyje 4.





- main\_template – pagrindinis puslapio šablonas;
- head\_tags – puslapio antraštės ir metaduomenų šablonas;
- hnavigation – dinamiškai kintančios meniu juostos šablonas;
- current\_navigation – lokalsios navigacijos šablonas;
- path\_navigation – naršymo kelio šablonas;
- select\_language – kalbų pasirinkimo šablonas;
- search – paieškos rezultatų šablonas;
- site\_map – svetainės žemėlapis šablonas;
- auth\_links – tarnybinio prisijungimo šablonas;
- actions\_box – administravimo įrankių bloko šablonas.

Šiuos standartinius šablonus galima rasti prisijungus prie „Zope“ serverio portal\_skins/RR direktorijoje. Norint juos taisyti, reikia pasirinkti konkretų šabloną ir paspausti mygtuką „Customize“. Atlikus šią procedūrą šablonas atsiranda „Custom“ direktorijoje. Šioje direktorijoje esančius failus jau galima redaguoti beveik bet kokiomis priemonėmis.

```

01 <metal:block define-macro="master">
02   <?xml version="1.0"?>
03   <!DOCTYPE html
04     PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
05     „http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd“>
06   <html>
07     <head>
08       <metal:block define-slot="base" />
09     </head>
10     <body>
11       [dizainerio HTML kodas]
12       <div metal:define-slot="header">
13
14       </div>
15       [dizainerio HTML kodas]
16       <div metal:define-slot="main">
17
18       </div>
19       [dizainerio HTML kodas]
20     </body>
21   </html>
22 </metal:block>

```

1 eilutė: <metal:block define-macro="master">

Šiomis kodo eilutėmis yra nurodoma, kad čia yra šablono vardu „master“ pradžia. Jau sukurtame HTML dokumente, šią eilutę reikia įterpti prieš <html> žymę, o dokumento pabaigoje uždaryti <metal:block> žymę. Šablono viduje nurodomos trys svetainės objektų turinio vietos (slots) – *base*, *header* ir *main*.

8 eilutė – **Base Slot:** <metal:block define-slot=“base“ />

Turinio valdymo sistema naudoja Base Slot'ą dinaminių nuorodų formavimui. Ši eilutė yra neredaguojama ir skirta TVS vidiniam naudojimui. Vienintelis reikalavimas, kad ši kodo eilutė būtų <head> sekcijoje.

12-14 Eilutės – **Header Slot:** <div metal:define-slot=“header“> ... </div>

Šioje vietoje objektai įterps savo antraštės tekstą. Pvz.: TVS objektas – katalogas šioje vietoje įterps savo pavadinimą ir katalogo aprašymą. Šios eilutės turi būti įterptos dizainerio nuožiūra, ten kur bus patogiau rodyti antraštes ir katalogų aprašymus.

16-18 Eilutės – **Main Slot:** <div metal:define-slot=“main“> ... </div>

Vieta, kurioje interneto svetainės objektai įterps savo turinį. Šis turinys bus generuojamas XHTML kodo pavidalu. Šios eilutės turi būti įterptos tokioje HTML dokumento vietoje, kurioje lankytoji bus pateikiami visi tekstai ir aprašymai. Grafinius failus rekomenduojama nusiųsti serveriui.

Norint įterpti išorinius failus, reikia patalpinti juos į „Custom“ direktoriją. Galima sukurti subdirektoriją, pvz. „images“ ir dėti failus ten.

Įtraukti paveikslus galima dviem būdais (tarkime, kad paveikslai sudėti direktorijoje .../custom/images):

Įprastas būdas:

```
<img src=“images/img.gif“ alt=“alternative“ title=“description“ width=“120“ height=“90“>
```

Priimtinesnis būdas:

```
<img src=“images/img.gif“ tal:replace=“structure here/images/img.gif“>
```

Antrasis būdas kompaktiškesnis. **Tal:replace** komanda pakeičia originalų **<img>** tagą į sugeneruotą, kuriame automatiškai įterpiami teisingi **width** ir **height** atributai, bei įdedami **alt** ir **title** atributai iš paveikslo metaduomenų. Vienintelis reikalavimas – paveikslas turi būti įvestas į turinio valdymo sistemos objektų duomenų bazę (naudojantis vartotojo sąsaja). Metaduomenis galima redaguoti naudojantis administravimo sąsaja, tai yra patogiu, jeigu prireikia juos vėliau pakeisti (pvz. Optimizuojant svetainės turinį paieškos mašinoms).

### 6.1.3 Išorinių CSS duomenų failų naudojimas

Įvairiuose šaltiniuose, vis dažniau rekomenduojama naudoti CSS stilių lenteles vietoje HTML komandų. Pagrindinės priežastys – paprastesnė realizacija, bei žymiai platesnės galimybės. Turinio

valdymo sistemoje yra numatytas platus CSS lentelių panaudojimas, bei jų prijungimas iš išorinių bylų.

CSS prijungimas realizuojamas dviem etapais:

- CSS lentelės bylos modifikavimas
- <link> kodinio žodžio įterpimas į HTML <head> sekciją.

Naudojant XHTML standartą, CSS failo pradžioje reikia įrašyti papildomą kodą:

```
/* <html-call „RESPONSE.setHeader(Content-Type,text/css)“> */. Šis kodas užtikrins HTTP Header parametro reikšmę – „Content-Type: text/css“. Taip pat reikia pakeisti <head> sekciją: <link rel="stylesheet" href="style.css" type="text/css"
```

```
tal:attributes="href here/style.css/absolute_url" />
```

**tal:attributes** parametras, interneto svetainės krovimosi metu generuos teisingą stilių lentelės adresą.

### 6.1.4 Dinaminių elementų įterpimas į šabloną

Kuriant aukštesnio lygio interneto svetaines, yra neapsiribojama tik statiškais dizaino elementais. Įterpus įvairius dinامينius elementus (pvz. Dinamiškai kintanti meniu juosta) bus pagerinta svetainės navigacija, nereikės kiekvieną kartą atnaujinti dokumentus taisyti HTML kodo. Šiame bus paaiškinta, kaip aprašyti ir įterpti šiuos elementus.

Dinaminio turinio įterpimui naudojami TAL XML vardų srities parametrai („atributai“) content ir replace:

#### **tal:content**

Žymė naudojama įdėti turinį į žymės vidų prieš sugeneruojant kodą interneto naršyklei.

```
<span class="title" tal:content="here/title" >The title</span> - įdės objekto antraštę <span> turinio („The title“) vietoje. Bus sugeneruotas kodas, panašus į tokį:
```

- <span class="title">Objekto pavadinimas</span>

#### **tal:replace**

Naudojamas pakeisti visą kodo eilutę (įskaitant pačią žymę, kurioje yra tal:replace atributas) į dinaminį turinį. Pavyzdžiui eilutė:

```
<span class="title" tal:replace="here/title"> The title</span>
```

bus pakeista tiesiog į:

```
Objekto pavadinimas
```

Šių atributų naudojimas yra labai patogus, nes HTML redaktoriai juos ignoruoja ir nerodo, o turinio valdymo sistema šias žymes pakeičia objekto duomenimis. Negalime tiksliai pasakyti, kuri **tal** parametras naudoti, nes tai priklauso nuo situacijos ir dizaineriui keliamos užduoties.

## 6.1.5 Objektų pavadinimų ir metaduomenų įterpimas

Norint į <head> sekciją įkelti dinامينius <title> ir <meta> elementus yra naudojamas šablonas **head\_tags**. Į pagrindinio (main\_template) šablono <head> sekciją, reikia įterpti šias eilutes:

```
<tal:block tal:replace="structure here/head_tags">
  <title>Title goes here</title>
  <meta name="description" content="document description" />
  <meta name="keywords" content="keyword list" />
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8;" />
</tal:block>
```

Naudojant šį kodą, dizaineris, redaguodamas dizaino šabloną HTML redaktoriumi matys statiškus elementus, o TVS juos pakeis į objekto <title> ir <meta> duomenis.

## 6.2 Dinaminiai elementai

Žemiau esančiuose skyreliuose bus pateikti pavyzdžiai kaip į šabloną įterpti dinامينius elementus.

### 6.2.1 Dinamiškai kintanti meniu juosta

Dinamiškai kintanti meniu juosta – tai nuorodos į interneto svetainės dokumentus. Keičiant dokumentų pavadinimus, sukuriant naujus dokumentus ar ištrinant, keisis ir meniu juosta. Norint įterpti dinamiškai kintančią meniu juostą, yra naudojamas toks kodas:

```
<div id="MainMenu" tal:content="structure here/hnavigation">
  Čia yra pagrindinis meniu
</div>
```

Dizaineris redaguodamas HTML kodą matys eilutę:

```
Čia yra pagrindinis meniu
```

O krovimosi metu, pagal dokumentų pavadinimus, turinio valdymo sistema sukurs HTML kodą, panašų į tokį:

```
<div id="MainMenu">
  <span class="menu-item">
    <a href="http://localhost/lt/naujienos">Naujienos</a>
  </span>
  <span class="menu-item">
    <a href="http://localhost /lt/index_html">RR TVS</a>
  </span>
  <span class="menu-item">
    <a href="localhost/lt/instrukcijos" class="current">Instrukcijos</a>
  </span>
</div>
```

Norint sukurti vertikalią meniu juostą (išdėstyti nuorodas stulpeliu), galima naudoti tokį CSS stilių:

```
#MainMenu .menu-item {
  display: block;}
```

Nuoroda į jau rodomą dokumentą naudoja “Current” stilių iš CSS lentelės. Keičiant Current stilių, galima efektingai paryškinti dokumento, kuriame yra lankytojas, pavadinimą.

Šių elementų užtenka, norint sukurti paprastą ir patogią interneto svetainės navigacijos sistemą. Norint realizuoti sudėtingesnę navigacijos meniu sprendimą, pakeitimai yra atliekami tiesiogiai **hnavigation** šablone.

## 6.2.2 Lokali navigacija

Norint lankytojui parodyti kuriame kataloge ar dokumente jis yra naršymo metu, bei šio katalogo submeniu, yra naudojamas **current\_navigation** šablonas:

```
<div id="CurrentMenu" tal:content="structure here/current_navigation">  
  Lokali navigacija  
</div>
```

Dizaineris redaguodamas HTML kodą matys šią eilutę:

Lokali navigacija

O krovimosi metu, pagal dokumento, kuriame yra lankytojas pavadinimą, TVS sukurs panašų

HTML kodą:

```
<div id="CurrentMenu">  
  <span class="menu-item">  
    <a href="http://localhost/lt/instrukcijos" class="current">Instrukcijos</a>  
  </span>  
</div>
```

## 6.2.3 Naršymo kelias

Šis elementas yra labai patogus lankytojams, nes jame matosi visos nuorodos, nuo pagrindinio („Home“) puslapio iki puslapio, kurį lankytojas peržiūrinėja. Šis dinaminis elementas gali būti įterptas naudojant **path\_navigation** šablona. Svetainės šablono HTML kode, norimoje vietoje, reikia įrašyti šias eilutes:

```
<span tal:content="structure here/path_navigation">  
  Naršymo kelias  
</span>
```

## 6.2.4 Kalbų pasirinkimas

Kalbų pasirinkimo meniu (atitinkamų šalių vėliavėlės su nuorodomis) yra įdedamas naudojant **select\_language** šablona. Svetainės šablono HTML kode, norimoje vietoje, reikia įrašyti šias eilutes:

```
<span tal:replace="structure here/select_language">  
  Kalbų pasirinkimas  
</span>
```

## 6.2.5 Paieškos laukas

Paieškos laukas yra sukuriamas naudojant specialiai paruoštą formos elementą. Paprasčiausios paieškos formos kodas:

```
<form action="search" method="get"
  tal:attributes="action string:${here/portal_url}/search">
  <input type="text" name="SearchableText" />
  <input type="submit" name="go" value="Search" i18n:attributes="value" />
</form>
```

**tal:attributes** parametras krovimosi metu pakeičia <form> **action** parametro reikšmę į turinio valdymo sistemos naudojamą kodą. Būtinai turi būti nurodytas įvedimo lauko **name** parametras – „SearchableText“. **i18n:attributes** parametras pakeičia <input> elemento „Search“ tekstą į vartotojo pasirinktą kalbą.

## 6.2.6 Svetainės planas

Viena iš labai patogių TVS funkcijų – dinamiškai kintančio interneto svetainės plano sukūrimas. Svetainės plane lankytoji pateiktos nuorodos į visus svetainės dokumentus. Nuorodos bus išdėliotos medžio principu.

Svetainės planas – tai specialus TVS dokumentas. Dizaineriui reikia patalpinti nuorodą į šį dokumentą svetainės šablone.

Galimos dviejų tipų nuorodos į svetainės planą:

- Pilnas svetainės planas visomis naudojamomis kalbomis;
- Svetainės planas, generuojamas kiekvienai iš naudojamų kalbų atskirai.

Nuoroda į pilną interneto svetainės planą, visomis kalboms, realizuojama naudojant **portal\_url** šablona, kuris gražina nuorodą į svetainės šaknį. Svetainės šablono HTML kode, norimoje vietoje, reikia įrašyti šias eilutes:

```
<a href="site_map" tal:attributes="href string:${here/portal_url}/site_map"
  i18n:translate="">Sitemap</a>
```

Nuoroda į svetainės planą, vienai kalbai, bus sukurta įrašius į svetainės šablono HTML kodo norimą vietą šias eilutes:

```
<a href="site_map" tal:define="local_root here/local_root"
  tal:attributes="href string:${local_root/absolute_url}/site_map"
  i18n:translate="">Sitemap</a>
```

## 6.2.7 Tarnybinis prisijungimas

Svetainės administratoriai galės patekti į prisijungimo prie TVS langą nuspaudę ant **tarnybinio prisijungimo** nuorodos. Ši nuoroda bus sukurta įrašius į interneto svetainės šablono HTML kodą šias eilutes:

```
<span tal:replace="structure here/auth_links">
  Tarnybinis prisijungimas
</span>
```

Svetainės lankytojai vietoje šio kodo matys nuorodą „Tarnybinis prisijungimas“. Prisijungę vartotojai matys savo prisijungimo vardą ir nuorodą „Išeiti“.

## 6.2.8 Administravimo įrankių blokas

Administratoriaus įrankių blokas yra įterpiamas naudojant šį kodą:

```
<div id="ContextActionsBox" tal:content="structure here/actions_box">
  Actions box placeholder
</div>
```

## 6.2.9 Pranešimai vartotojui

Vartotojui atliekant pakeitimus interneto svetainėje, TVS pranešimais informuos apie vartotojo veiksmų rezultatus. Pranešimas vartotojui yra perduodamas naudojant dinaminį kintamąjį **portal\_status\_message**. Interneto svetainės šablone reikia įrašyti šias eilutes:

```
<p id="DesktopStatusBar"
  tal:condition="request/portal_status_message|nothing"
  tal:content="request/portal_status_message">Status message.</p>
```

**tal:condition** parametras užtikrins, kad kintamojo reikšmė bus rodoma tik kuomet, kai tas kintamasis apibrėžtas, t.y. atsiranda naujas pranešimas vartotojui. Pranešimas apipavidalinimas įterpiant **#DesktopStatusBar** pranešimų stilių į naudojamą CSS lentelę.

## 6.2.10 „Corp” kalendorius

„Corp” kalendorius – tai paprastas, bet labai efektyvus „Python” produktas. Šis produktas patogus ir naudingas nedidelėms įmonėms. Palengvina įmonių darbuotojų tarpusavio bendravimą interneto naršyklės pagalba. Produkto integravimas:

- produkto katalogą įrašome i direktorija /Products/;
- perkrauname „Zope” serverį;
- sukuriame objektą „CorpCalendar”.



## 7. Išvados

1. Atlikus turinio valdymo sistemų internete analizę Lietuvos ir pasaulio mastu, prieita išvada suprojektuoti ir realizuoti turinio valdymo sistemą su savybėmis specifikuotomis ir aprašytomis analizės dalyje.
2. Turinio valdymo sistemų vartotojų poreikių analizė parodė, kad būtina turinio valdymo sistemą išplėsti. Tai yra į bazinę turinio valdymo sistemos versiją reikia integruoti tokius produktus kaip e-commerce, CRM, B2C ir daugelį kitų produktų.
3. Renkantis kūrimo platformą, svarstyta keletas alternatyvų: PHP ir Zope+Python technologijos. Galutinis pasirinkimas buvo Zope+Python technologijos dėl plataus kūrimo, palaikymo priemonių ir komponentų pasirinkimo, lengvai pritaikomos atvirojo kodo koncepcijos, nepriklausomumo nuo platformos. Taip pat Zope+Python platforma buvo pasirinkta dar ir dėl to, kad ji leidžia patogiai kurti nedideles interneto svetaines, ir turi rimtų privalumų, kai reikia dirbti su didesniais projektais. Lyginant su PHP, Zope+Python turi geresnę vidinę logiką ir integruotus įrankius, padedančius ne tik efektyviau dirbti, bet taip pat ir lengviau prižiūrėti, suprasti jau sukurtą kodą.
4. Projektinėje dalyje buvo apibrėžtas kuriamo produkto tikslas, bei taikymo sritis. Taip pat detaliai aprašyti vartotojų panaudojimo atvejai ir sistemos elgsenos modelis.
5. Remiantis analitinėje dalyje suformuluotais reikalavimais ir priimtais projektiniais sprendimais suprojektuota ir realizuota turinio valdymo sistema, kuri pasižymi efektyviu puslapių turinio ir dizaino redagavimu ir administravimu. Sistema lengvai įsisavinama vartotojų. Puslapiai teisingai formuojami visose naršyklėse. Sukurtų internetinių svetainių struktūra suderinta su žinomiausiomis paieškos sistemomis. Išplečiant bazines TVS savybes buvo integruoti įrankiai „CorpCalendar“ ir „ButtonsBuilder“, kurie palengvina tiek dizainerio darbą, tiek mažų įmonių darbuotojų tarpusavio bendravimą interneto naršyklės pagalba. Taip pat sukurtas pagrindinis šablonas main\_template, kuris apsprendžia standartinės svetainės, kuriamos su realizuota TVS, vaizdą.
6. Atliktas eksperimentinis turinio valdymo sistemos tyrimas rodo, kad savybės ir reikalavimai, kurie buvo apibrėžti realizuojamai turinio valdymo sistemai, įgyvendintos. Sistemos įdiegimas paprastas, turinio administravimas nesudėtingas net ir eiliniam vartotojui.
7. Vykdamas galimus ateities išplėtimus, pasirinkta turinio valdymo sistemos modelio realizavimo platforma leis nesunkiai praplėsti TVS funkcionalumą.

## 8. Literatūra

1. The Zope Book On-line. [Žiūrėta 2005 06 15] Prieiga per internetą <http://www.zope.org/Documentation/Books/ZopeBook/>
2. Welcome to opensourceCMS. [Žiūrėta 2005 05 12] Prieiga per internetą <http://www.opensourcecms.com/>
3. Using Relation Data Base With Zope and Plone. [Žiūrėta 2005 04 19] Prieiga per internetą <http://plone.org/Members/pupq/reldb/view?searchterm=change%20ZODB%20to%20relational%20DB>
4. Plone: A user-friendly and powerful open source Content Management System [Žiūrėta 2005 03 02] Prieiga per internetą <http://plone.org/>
5. Model.Logical.ZODB.Packing and Garbage Collection. [Žiūrėta 2005 06 01] Prieiga per internetą <http://www.zope.org/Documentation/Developer/Models/ZODB/>
6. Turinio valdymo sistema InterTVS. [Žiūrėta 2005 05 24] Prieiga per internetą <http://www.intertvs.lt/>
7. SAGE internetas turinio valdymo sistema. [Žiūrėta 2005 05 24] Prieiga per internetą <http://www.sage.lt/page.php?pId=128&lang=lt>
8. Optimal Site Content Management System. [Žiūrėta 2005 05 24] Prieiga per internetą <http://www.optimalsite.com/>
9. Zet Interneto sprendimai. [Žiūrėta 2005 05 24] Prieiga per internetą <http://www.zet.lt/puslapiukurimas.html?PHPSESSID=751e3d893d7b0275dae2aaca4ca99fb4>
10. Abi GROUP turinio valdymo sistema. [Žiūrėta 2005 05 15] Prieiga per internetą [http://www.abigroup.lt/?p=turinio\\_valdymo\\_sistema](http://www.abigroup.lt/?p=turinio_valdymo_sistema)
11. CMS Matrix. [Žiūrėta 2005 06 21] Prieiga per internetą <http://www.cmsmatrix.org>
12. XML-RPC. [Žiūrėta 2005 04 10] Prieiga per internetą <http://www.xml-rpc.org>
13. Ramanauskas A., Svetainės turinio valdymas – Interneto rinkodaros įrankis (II), [žiūrėta 2005 11 24]. Prieiga per internetą <http://www.sitesupra.lt>.
14. Kappe F.: “XML and Hyperwave Information Server”, techniniai užrašai, 1999 [Žiūrėta 2005 04 23]. Prieiga per internetą <http://www.hyperwave.com>
15. Reynolds J., Kaur A.: “Content Management”, [Žiūrėta 2004 04 15]. Prieiga per internetą <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/ecommerce/maintain/operate/contmgmt.asp>

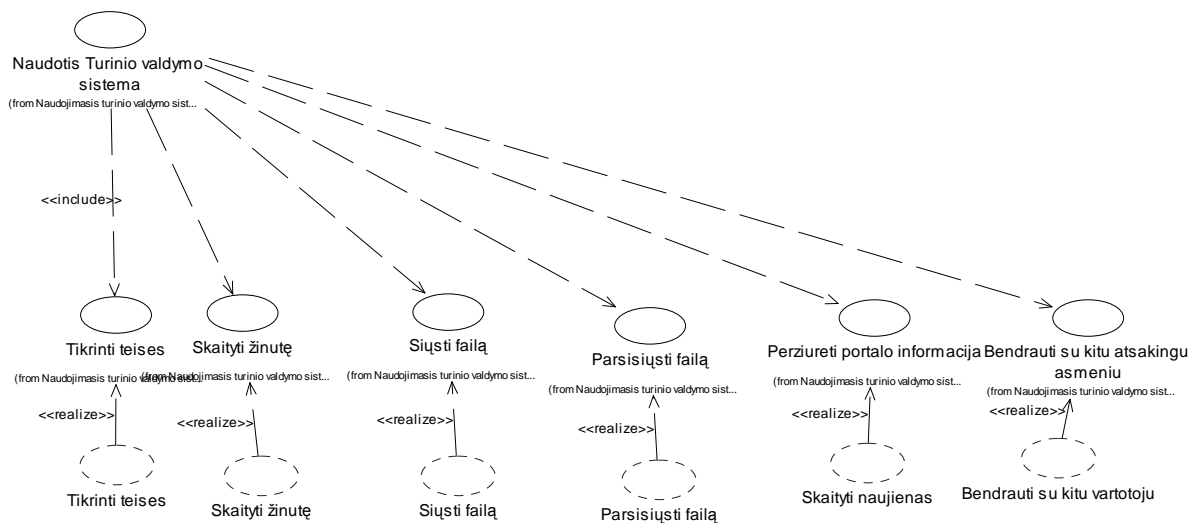
16. Corp Calendar is a simple little calendar application suitable for small groups or companies.[Žiūrėta 2005 12 02]. Prieiga per internetą <http://www.zope.org/Members/malikz/CorpCalendar>
17. EasyWeb sistema. [Žiūrėta 2005 05 12]. Prieiga per internetą <http://www.easyweb.lt/>

## 9. Terminų ir santrumpų žodynas

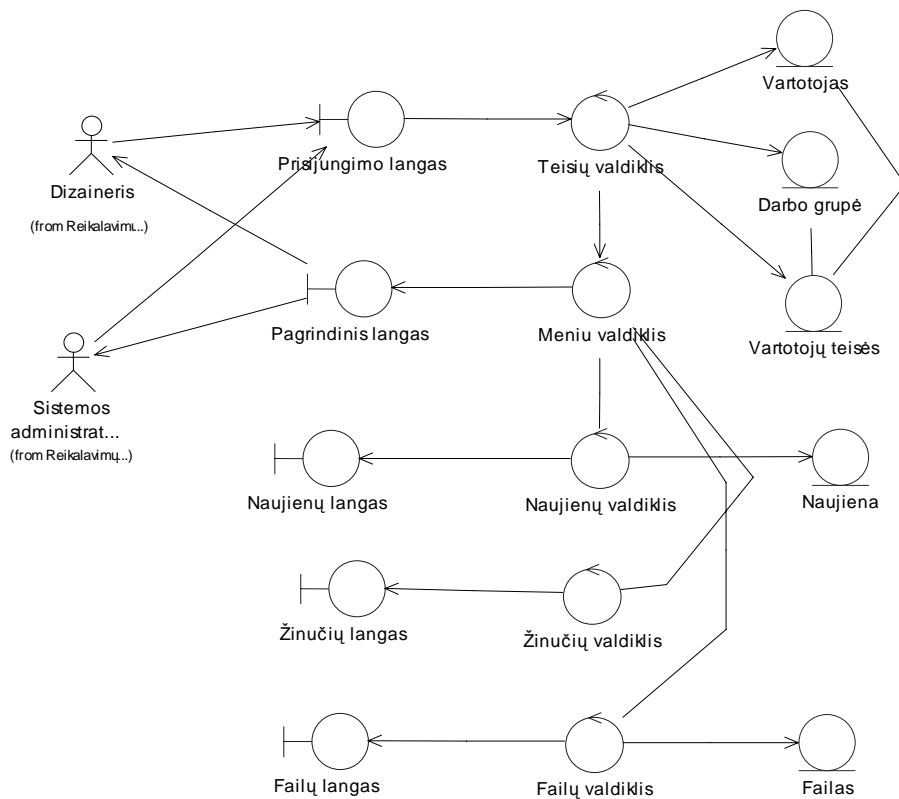
Terminas	Apibrėžimas
DBVS	duomenų bazių valdymo sistema
ZODB	objektinė Zope duomenų bazė
Zope	Web serveris
ZTP	Zope Page Template
Framework	Karkasas. Taip apibūdinamas sprendimas pateikiantis architektūrinį sprendimą (design template)
Naršyklė	Programa, skirta peržiūrėti interneto tiekiamus hipertekstinius dokumentus.
HTTP	Protokolas, skirtas perduoti duomenis internete.
HTML	Hipertekstinių dokumentų tipas.
Administratorius	Bet kokios sistemos vartotojas, turintis visas ar dalines teises keisti sistemos konfigūraciją ir informaciją
OS	Operacinė sistema
DB	Duomenų bazė
IIS	Informacinis interneto serveris ( <i>Internet Information Sever</i> ).
CSS	Stiliaus šablonas
Tags	Tagai
Serverio komponentai	Komponentai talpinami tarnybinėje stotyje, sąveikaujantys su kliento kompiuteriu pagal daugialypės architektūros principus.
I18N	Internaciolizacija – metodika, kaip dinamiškai išvesti pranešimus vartotojui pagal kompiuterio lokalę.

# Priedas1

## Naudojimosi sistema realizacija

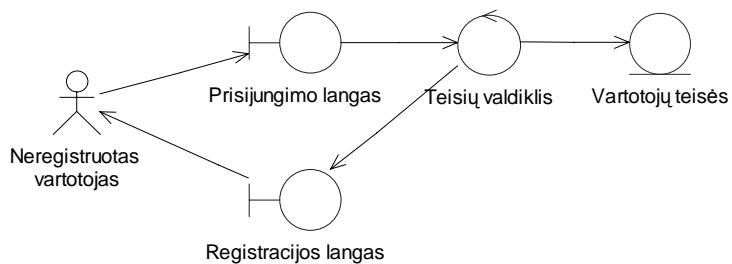


19 pav. Trasų diagrama



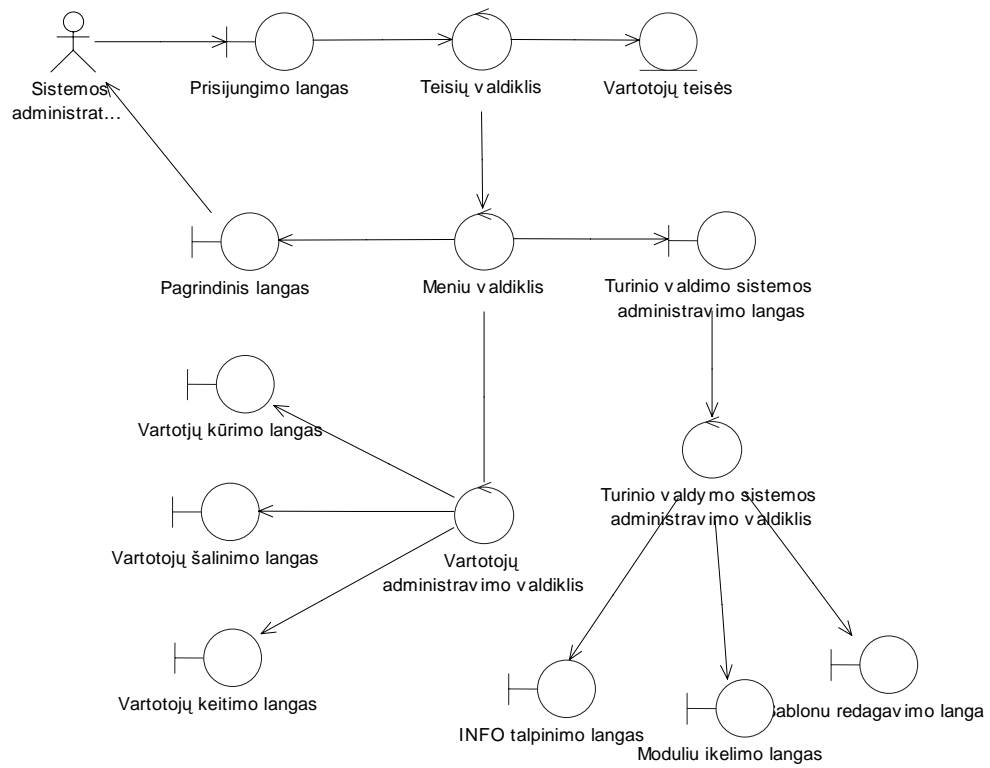
**20 pav. Naudojimosi sistema trasų diagrama**

Naudojimosi sistema realizacija



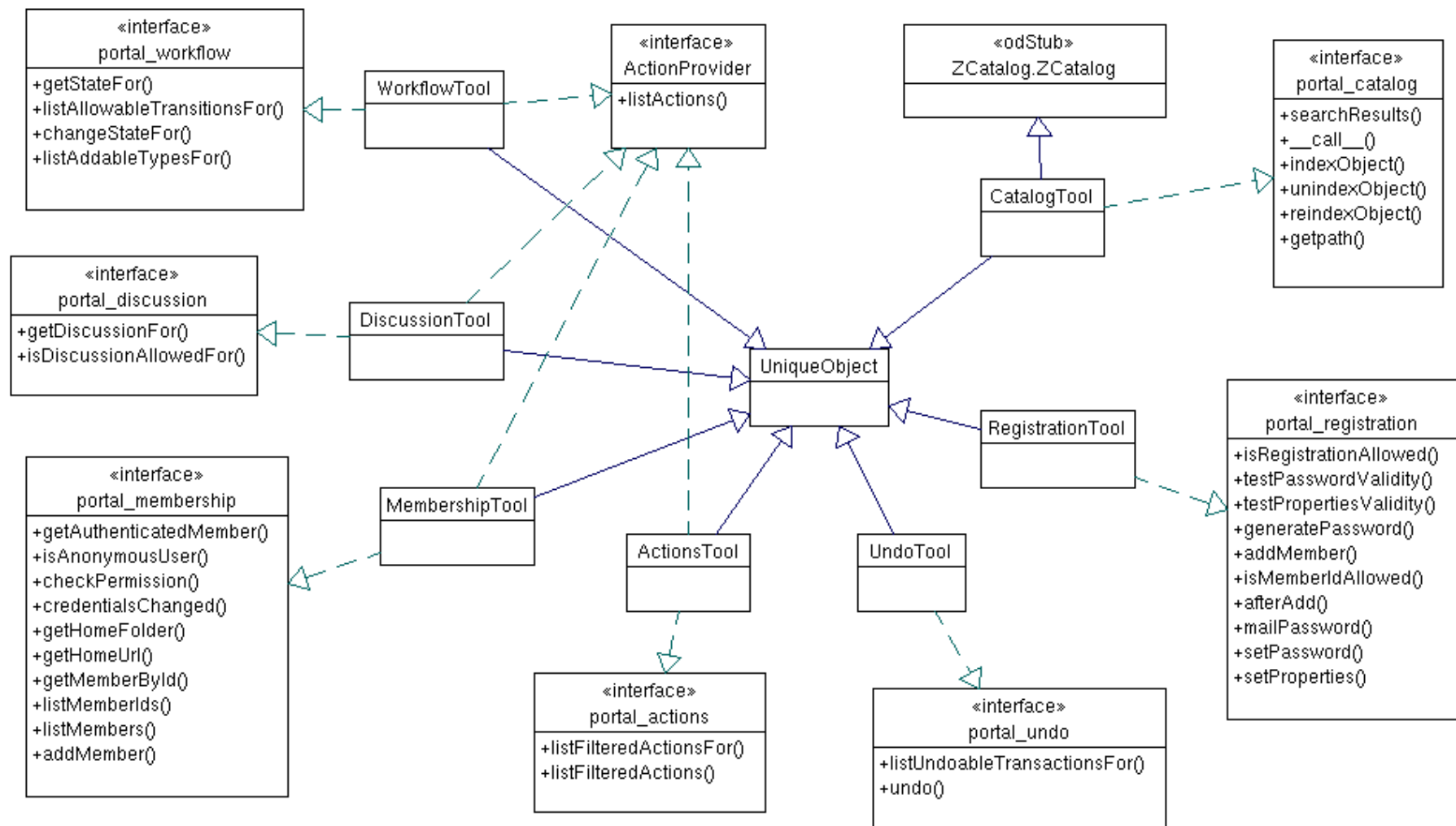
**21 pav. Naudojimosi sistema trasų diagrama**

Sistemos administravimo realizacija



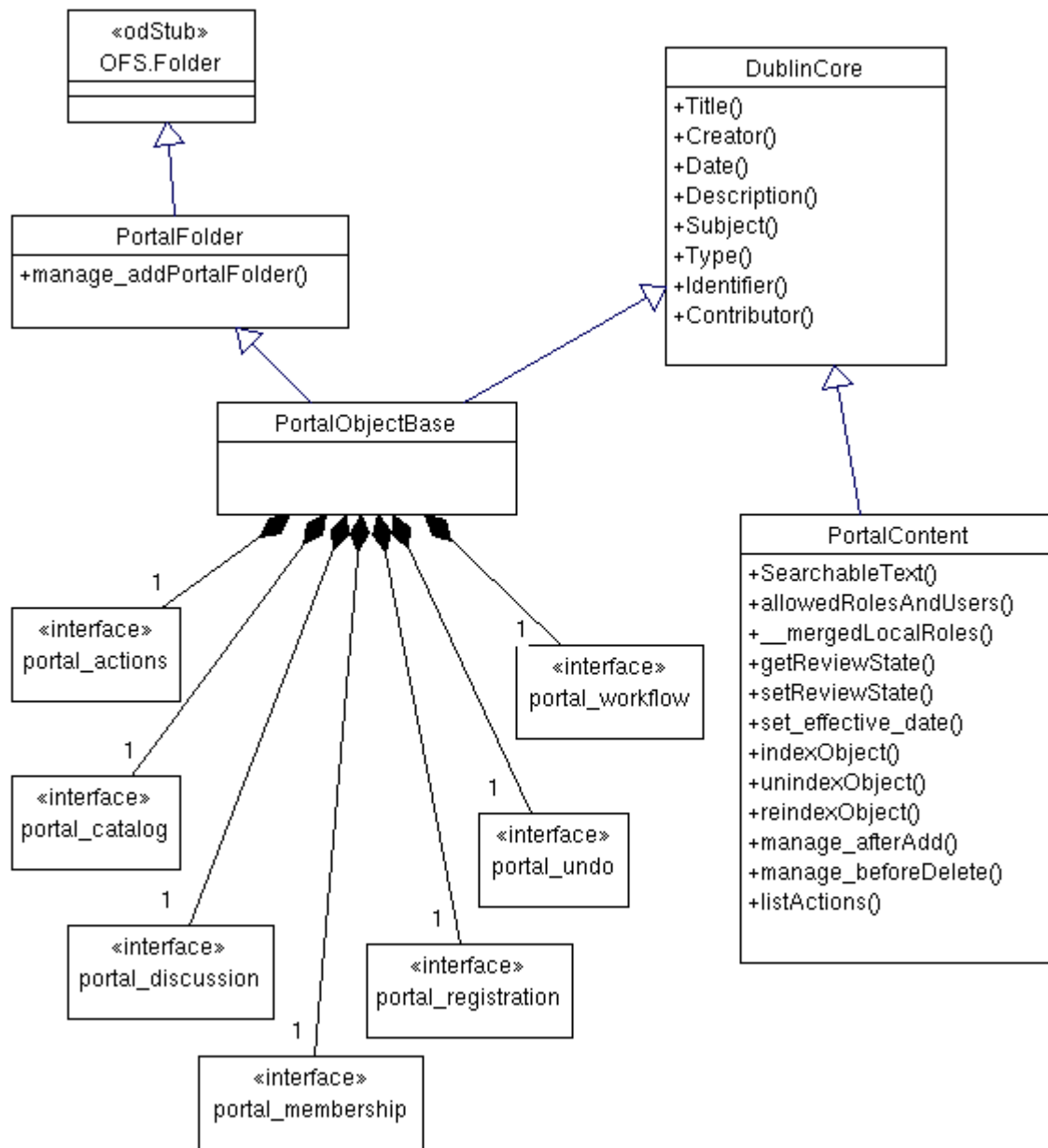
**22 pav. Sistemos administravimo trasų diagrama**

## Priedas 2



23 pav. Karkaso įrankių diagrama





24 pav. Karkaso objektų diagrama

4 lentelė. Koduotės poveikis skirtingų simbolių atvaizdavimui

UTF-8	Windows-1257	Iso-8859-1
EN - qwerty	EN - qwerty	EN - qwerty
LT - ačėėįš	LT - ačėėįš	LT - àèëáđ
RU - йцукен	RU - ??????	RU - ??????