VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

Nerijus Bučinskis

ELEKTRONINIŲ PASLAUGŲ NAUDOTOJO SĄSAJOS KŪRIMAS EPOLICIJA.LT PORTALE

DEVELOPMENT OF USER INTERFACE FOR ELECTRONIC SERVICES IN EPOLICIJA.LT PORTAL

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 621E14004
Duomenų gavybos technologijų specializacija
Informatikos inžinerijos studijų kryptis

Vilnius, 2013

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU
Katedros vedėjas

_____

(Parašas)

_____

(Vardas, pavardė)

_____

(Data)

Nerijus Bučinskis

ELEKTRONINIŲ PASLAUGŲ NAUDOTOJO SĄSAJOS KŪRIMAS EPOLICIJA.LT PORTALE

DEVELOPMENT OF USER INTERFACE FOR ELECTRONIC SERVICES IN EPOLICIJA.LT PORTAL

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 621E14004

Duomenų gavybos technologijų specializacija

Informatikos inžinerijos studijų kryptis

Vadovas _____ _____ _____

(Moksl. laipsnis/pedag. vardas, vardas, pavardė) (Parašas) (Data)

Lietuvių kalbos konsultantas _____ _____ _____

(Moksl. laipsnis/pedag. vardas, vardas, pavardė) (Parašas) (Data)

Vilnius, 2013

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių technologijų katedra

ISBN ISSN
Egz. sk. 2
Data …....-….-….

Informacinių technologijų studijų programos baigiamasis magistro darbas

Pavadinimas: Elektroninių paslaugų naudotojo sąsajos kūrimas epolicija.lt portale

Autorius **Nerijus Bučinskis** Vadovas doc. dr. **Egidijus Ostašius**

Kalba

X	Lietuvių
	Užsienio

Anotacija

Magistro darbo tema – „Elektroninių paslaugų naudotojo sąsajos kūrimas epolicija.lt portale“. Jame nagrinėjamas policijos elektroninių paslaugų portalo kūrimo procesas, renkamos tinkamiausios technologijos sukurti lanksčią, modernią ir patogią naudotojo sąsają, kuriant ir užsakant elektronines policijos paslaugas.

Pagrindinis tikslas yra sukurti pažangų paslaugų administravimo modulį, kuriuo sistemos administratorius galėtų kurti ir administruoti policijos elektroninių paslaugų duomenų priėmimo formas be programavimo žinių.

Įgyvendinant elektroninių paslaugų naudotojo sąsajos kūrimą epolicija.lt portale, buvo atlikta technologijų analizė, parinktos tinkamiausios technologijos kurti modernias taikomąsias programas. Jų pagalba buvo sukurtas universalus formų konstravimo komponentas, pritaikomas sistemose, kurioms reikalinga administruoti paslaugų formas.

Darbą sudaro: įvadas, epolicija.lt analizė, epolicija.lt naudotojo sąsajos realizacija, rezultatai ir išvados.

Darbo apimtis - 48 puslapiai be priedų, 23 paveikslai, 17 lentelių ir 16 informacijos šaltinių.

Prasminiai žodžiai: internetinė taikomoji programa, PEPS, Liferay, ZK, elektroninė paslauga, paslaugos forma, policija

Vilnius Gediminas Technical University
Fundamental Sciences faculty
Information Technologies department

ISBN ISSN
Copies No. 2
Date …....-….-….

Information technology study program master thesis.

Title: Development of user interface for electronic services in epolicija.lt portal

Author **Nerijus Bučinskis** Academic supervisor doc. dr. **Egidijus Ostašius**

Thesis language

X	Lithuanian
	Foreign

Annotation

Theme of final master thesis – „Development of user interface for electronic services in epolicija.lt portal“. This work contains analysis for creating police electronic services portal and picking the best technologies to create verbose, modern and user friendly user interface for creating and using police electronic services.

Main goal is to create modern service administration module, which could be used by system administrator to create and manage data input forms for police services, without any programming knowledge.

During the development of user interface for electronic services in epolicija.lt portal, technology analysis was performed, which highlighted most suitable software for creating modern web applications. Using these technologies, universal form generation component was created, which could be used in other systems, which require service form administration.

Thesis consists of introduction, analysis of epolicija.lt, development of user interface in epolicija.lt, results and conclusions.

Thesis consists of 48 pages without appendixes, 23 pictures, 17 tables and 16 information sources.

Keywords: WEB application, PEPS, Liferay, ZK, electronic service, service form, police

Vilniaus Gedimino technikos universiteto egzaminų, sesijų ir baigiamųjų darbų rengimo bei gynimo organizavimo tvarkos aprašo 2011-2012 m. m.

1 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Nerijus Bučinskis, 20073135

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Informacinės technologijos, DGTfm-11

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO) SĄŽININGUMO DEKLARACIJA

2013 m. gegužės 31 d.

Patvirtinu, kad mano baigiamasis darbas tema „Elektroninių paslaugų naudotojo sąsajos kūrimas epolicija.lt portale" patvirtintas 2012 m. spalio 24 d. dekano potvarkiu Nr. 375fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Parenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: doc. dr. Angelė Kaulakienė. Mano darbo vadovas dr. Egidijus Ostašius.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas) Nerijus Bučinskis

(Vardas ir pavardė)

TURINYS

1. ĮVADAS ... 10
2. EPOLICIJA.LT ANALIZĖ ... 14
2.1. Panaudos atvejai ... 16
2.1.1. Pranešimo policijai teikimas ... 19
2.1.2. Pareiškimo, skundo ar prašymo policijai teikimas ... 20
2.2. Asmenų identifikavimas ir tapatybės nustatymas ... 21
2.3. El. dokumento kūrimas ir pasirašymas ... 21
2.4. Duomenų bazės modelis ... 22
2.4.1. Lentelė „SERVICE_REQUEST“ ... 23
2.4.2. Lentelė „SERVICE_DEFINITION“ ... 24
2.4.3. Lentelė „SERVICE_PROVIDER“ ... 25
2.4.4. Lentelė „SERVICE_FIELD“ ... 25
2.4.5. Lentelė „SERVICE_FORM“ ... 25
2.4.6. Lentelė „FORM_LAYOUT“ ... 26
2.4.7. Lentelė „LAYOUT_NODE“ ... 26
2.5. Technologijų analizė ... 27
2.5.1. Portalo pagrindas ... 28
2.5.2. Programiniai karkasai ... 31
2.5.3. Google žemėlapių paslaugos ... 34
3. EPOLICIJA.LT NAUDOTOJO SĄSAJOS REALIZACIJA ... 35
3.1. Sistemos administratoriaus dalis ... 35
3.1.1. Prisijungimas prie administravimo sąsajos ... 36
3.1.2. Pradinis portalo administravimo sąsajos puslapis ... 37
3.1.3. Paslaugų administravimo puslapis ... 38
3.1.4. Naujos paslaugų sistemos ir paslaugos sukūrimas, keitimas ir šalinimas ... 40
3.1.5. Paslaugos duomenų įvedimo formos laukų šablonai ... 42
3.1.6. Paslaugos užsakymo duomenų formos sukūrimas, keitimas ir šalinimas ... 43
3.1.7. Formos nuostačių langas ... 43
3.1.8. Formos komponentų parametrai ... 45
3.1.9. Formos peržiūra ... 49
3.2. Naudotojo dalis ... 51
3.2.1. Prisijungimas prie portalo ... 51
3.2.2. Visos epolicija.lt paslaugos ... 51
3.2.3. Pranešimas policijai ... 52
3.2.4. Naudotojo duomenys ... 52
3.2.5. Naudotojo pranešimai ... 53
4. REZULTATAI IR IŠVADOS ... 55

INFORMACIJOS ŠALTINIŲ IR LITERATŪROS SĄRAŠAS 57
PRIEDAI 58

Paveikslų sąrašas

1 pav. Supaprastinta PEPS pranešimų valdymo schema 15
2 pav. PEPS architektūra 16
3 pav. Bendra PEPS portalo panaudos atvejų diagrama 17
4 pav. Pranešimo teikimo policijai diagrama 19
5 pav. Pareiškimo ar skundo teikimo diagrama 20
6 pav. ADOC generavimo ir pasirašymo diagrama 22
7 pav. Esybių (lentelių) sąryšio schema 23
8 pav. Paslaugos formos kūrimas 36
9 pav. Prisijungimo forma 37
10 pav. Pradinis epolicija.lt portalo administravimo sąsajos puslapis 38
11 pav. Įrankių juosta 38
12 pav. Paslaugų administravimo langas 39
13 pav. Paslaugos sistema 40
14 pav. Paslaugos duomenys 41
15 pav. Paslaugos duomenų įvedimo formos laukų šablonai 42
16 pav. Paslaugos užsakymo duomenų formos 43
17 pav. Paslaugos duomenų įvedimo forma 44
18 pav. Formos komponento kontekstinis meniu 48
19 pav. Komponento etikečių redagavimas: a) įvedimo langas b) rezultatas 48
20 pav. Paslaugos duomenų formos struktūros medis 49
21 pav. Paslaugos duomenų forma peržiūros režimu 50
22 pav. Teikiamų paslaugų sąrašas su vertinimo galimybe 52
23 pav. Prisijungusio naudotojo užsakytos paslaugos 54

Lentelių sąrašas

1 lentelė. Panaudos atvejai 18
2 lentelė. Pranešimo teikimo policijai aprašymas 19
3 lentelė. Pareiškimo ar skundo teikimo aprašymas 20
4 lentelė. Duomenų bazės lentelės „SERVICE_REQUEST“ laukai 23
5 lentelė. Duomenų bazės lentelės „SERVICE_DEFINITION“ laukai 24
6 lentelė. Duomenų bazės lentelės „SERVICE_PROVIDER“ laukai 25
7 lentelė. Duomenų bazės lentelės „SERVICE_FIELD“ laukai 25
8 lentelė. Duomenų bazės lentelės „SERVICE_FORM“ laukai 26
9 lentelė. Duomenų bazės lentelės „FORM_LAYOUT“ laukai 26
10 lentelė. Duomenų bazės lentelės „LAYOUT_NODE“ laukai 27
11 lentelė. Internetinių portalų palyginimas 29
12 lentelė. Programinių karkasų palyginimas 32
13 lentelė. Paslaugos sistemos administravimo mygtukai 40
14 lentelė. Paslaugos duomenų laukų aprašymas 41
15 lentelė. Šablonų laukų aprašymai 42
16 lentelė. Paslaugos duomenų įvedimo formos komponentai 45
17 lentelė. Formos komponentų parametrai 46

Santrumpų sąrašas

WEB – interneto tinklas (angl. *World Wide Web*)

AJAX – asinchroninis JavaScript ir XML (angl. *Asynchronous JavaScript And XML*)

HTTP – žiniatinklio protokolas (angl. *Hyper Text Transfer Protocol*)

HTML – hiperteksto žymėjimo kalba (angl. *Hyper Text Markup Language*)

WYSIWYG – „ką matai tą ir gauni“ (angl. *What You See Is What You Get*)

MVC – modelio-vaizdo-valdiklio architektūra (angl. *Model-View-Controller*)

MVVM – modelio-vaizdo-vaizdo modelio architektūra (angl. *Model-View-View Model*)

XML – žymėjimo kalba (angl. *Extensible Markup Language*)

DOC – „Microsoft Word“ dokumentų formatas

JMS – standartizuota „Java API“ sąsaja, skirta sistemų komunikacijai (angl. *Java Message Service*)

GR - LR gyventojų registras

PRĮR - policijos registruojamų įvykių registras

PPP - policijos personalo posistemis

ADMIN - naudotojų prieigos teisių valdymo sistema

AUDIT - naudotojų veiksmų kontrolės sistema

PUVS - pranešimų, užduočių valdymo sistema

ATPEĮR - administracinių teisės pažeidimų ir eismo įvykių registras

ĮKTAŽR - įtariamų, kaltinamų, teistų asmenų žinybinis registras

JAR - juridinių asmenų registras

NVŽR - nusikalstamų veikų žinybinis registras

PEPS - policijos elektroninių paslaugų sistema

VIISP - valstybės informacinių išteklių sąveikumo platforma

SMS - trumposios žinutės (angl. *Short Message Service*)

SIGNA - elektroninio parašo komponentas

1. ĮVADAS

Paslaugų, kuriomis pasinaudoti tekdavo keliauti į konkrečią įstaigą, užpildyti daugybę popierinių formų, lieka vis mažiau – visa tai keičia elektroninės sistemos. Policija yra viena iš tų sričių, kurios teikia tik labai mažą dalį elektroninių paslaugų, tačiau situaciją bando keisti *epolicija.lt* elektroninių paslaugų portalas (toliau - PEPS portalas). Jame siekiama suteikti visą paslaugų paketą gyventojams, kad nereikėtų išeiti iš namų. Pradinis PEPS tikslas yra sukurti sistemą kurioje būtų dvi pagrindinės elektroninės paslaugas:

- informacijos apie nusikalstamą veiką, eismo įvykį ar kitus teisės pažeidimus priėmimas;
- skundo, pareiškimo, pranešimo apie nusikalstamą veiką priėmimas.

Papildomai reikalinga sudaryti sąlygas diegti naujas paslaugas, modifikuoti esamas paslaugas bei susijusius procesus, užduotis, veiklos taisykles. PEPS turi būti suprojektuota taip, kad ateityje būtų galima be didelių papildomų programavimo darbų įdiegti šias viešąsias paslaugas:

- skundų, prašymų, susijusių su eismo priežiūra priėmimas;
- juridinių ir fizinių prašymų dėl leidimo įsigyti ginklą išdavimo priėmimas;
- juridinių ir fizinių asmenų prašymų dėl išankstinio sutikimo įvežti ginklus į Lietuvos Respublikos išdavimo priėmimas;
- juridinių asmenų prašymų dėl leidimų įvežti, išvežti, gabenti tranzitu civilines pirotechnikos priemones priėmimas;
- juridinių ir fizinių asmenų prašymų dėl leidimų užsakyti antspaudo su valstybės herbu pagaminimą priėmimas;
- juridinių asmenų prašymų dėl leidimų nešiotis juridinių asmenų ginklus, apsaugos darbuotojų ir inkasatorių pažymėjimų priėmimas;
- juridinių asmenų prašymų dėl leidimų įvežti, išvežti ar gabenti ginklus, šaudmenis tranzitu priėmimas;
- fizinių ir juridinių asmenų prašymų dėl leidimų įvežti (išvežti) ginklus priėmimas;
- juridinių ir fizinių asmenų prašymų dėl leidimų vežti ginklus į Europos Sąjungos valstybes priėmimas;
- fizinių asmenų prašymų dėl Europos šaunamojo ginklo leidimų išdavimo priėmimas;
- fizinių asmenų pranešimų apie ginklo laikymo sąlygas priėmimas;
- prašymų dėl leidimų įrengti žybčiojančius švyturėlius išdavimo priėmimas;
- prašymų dėl leidimų transporto priemonėje naudoti tamsintus stiklus išdavimo priėmimas;
- prašymų dėl inkasavimo pažymėjimų transporto priemonėms išdavimo priėmimas;

- prašymų dėl vienkartinių leidimų nuvažiuoti automobiliu į techninę apžiūrą išdavimo priėmimas;
- prašymų dėl registracijos derinti projektus, susijusius su kelio ženklais, priėmimas;
- prašymų dėl registracijos derinti statybos projektų, susijusių su eismo saugumu, priėmimas;
- prašymų dėl pažymų apie asmens padarytus eismo įvykius išdavimo priėmimas;
- prašymų dėl dokumentų, esančių policijos įstaigose (baudžiamosiose, administracinėse bylose ir kt.) kopijavimo ir teikimo suinteresuotiems asmenims priėmimas;
- prašymų dėl informacijos teikimo, pažymų apie asmens padarytus teisėtvarkos pažeidimus išdavimo iš policijos tvarkomų žinybinių registrų priėmimas;
- prašymų dėl pažymų apie asmens sulaikymą, suėmimą išdavimo priėmimas;
- prašymų dėl DNR, tėvystės nustatymo tyrimų ir ekspertizių priėmimas;
- prašymų dėl dokumentų blankų ir rekvizitų tyrimų ir ekspertizių priėmimas;
- prašymų dėl rašysenos tyrimų ir ekspertizių priėmimas;
- prašymų dėl transporto priemonių tyrimų ir ekspertizių priėmimas bei jų atlikimas;
- prašymų dėl balistinių tyrimų ir ekspertizių priėmimas ir jų atlikimas;
- prašymų dėl Lietuvos Respublikos pasų ar tapatybės kortelių išdavimo priėmimas;
- prašymų dėl leidimų gyventi Lietuvos Respublikos (nuolat ar laikinai) priėmimas;
- prašymų dėl kvietimų užsieniečiams laikinai atvykti į Lietuvos Respublikos priėmimas;
- prašymų dėl laikinų piliečio pažymėjimų išdavimo priėmimas;
- prašymų dėl kelionės dokumentų išdavimo priėmimas;
- prašymo dėl pažymos dėl prarastų asmens dokumentų išdavimo priėmimas;
- prašymo dėl paskutinės mirusio asmens gyvenamosios vietos priėmimas [10].

Sistemai realizuoti galimi du būdai:

1. Tradicinis programavimas. Visas funkcionalumas realizuojamas programiniu būdu, visos paslaugos įtraukiamos į projektuojamą sistemą. Tokiu būdu visi pakeitimai yra negalimi be programavimo žinių bei kiekvienas atnaujinimas, pavyzdžiui, nauja paslauga, reikalauja visiško sistemos atnaujinimo.
2. Sukurti sistemos paslaugą kaip bendrinį objektą, kurį būtų galima pridėti, ištrinti, pakeisti. Tokiu būdu visą sistemos priežiūrą galima būtų perleisti administratoriui, kuris netgi neturėdamas programavimo žinių galėtų atlikti visus reikalingus veiksmus.

Sprendimo pasirinkimas yra labai aktualus ir finansiniu atžvilgiu. Pirmuoju būdu išlaidos nesibaigtų kūrimo pabaigoje, o vėl atsirastų sistemą kaskart atnaujinant, ypač kai reikalinga nauja paslauga. Renkantis antrąjį sprendimo būdą finansinė problema sprendžiama, tačiau vienas iš sudėtingiausių iššūkių yra sukurti naudotojo sąsajos administravimo modulį. Asmuo, prižiūrintis sistemą, turėtų galėti sukonstruoti duomenų priėmimo formą, pridėti reikalingus komponentus, nustatyti priimamų duomenų tipus ir savybes.

Norint užtikrinti tokį lankstumą, reikalingos modernios technologijos, kurios leistų įgyvendinti tokį sudėtingą uždavinį. Kuriant patogią naudotojo sąsają, reikia numatyti sistemos naudotojų poreikius bei sukurti tokią sistemą, kuri leistų pasirinkti vietą žemėlapyje, įkelti vaizdinį ar garsinį failą, įgalinti automatinį reikšmių užpildymą, atlikti dokumentų pasirašymą elektroniniu parašu ir kitus veiksmus, neatsiejamus su moderniomis taikomosiomis programomis.

Realizavus paslaugų administravimą be programuotojo įsikišimo, principus ir programinę įrangą galima būtų pritaikyti ne tik PEPS portale, bet ir kitose srityse, kuriose reikalingas paslaugų kūrimas ir valdymas.

Darbo tikslas. Išanalizuoti PEPS reikalavimus, susijusius su paslaugos formų kūrimu, parinkti tinkamiausias technologijas bei realizuoti modernią valdymo sąsają dinamiškoms paslaugų formoms.

Projekto metodikos aprašymas. Analitinėje dalyje aprašomi panaudos atvejai, susiję su paslaugos naudojimu PEPS portale, išanalizuojamos bei parenkamos tinkamiausios technologijos, skirtos moderniam ir viešai prieinamam portalui sukurti. Praktinėje dalyje detalizuojama sukurta naudotojo bei administratoriaus naudotojo sąsaja, išanalizuojamas administratoriaus paslaugų valdymo modulio funkcionalumas ir naudojimas, išskiriamos paslaugų naudotojo galimybės.

Darbo uždaviniai

- Išanalizuoti panaudos atvejus, taikomus paslaugų užsakyme bei esamas technologijas, leidžiančias kurti modernias internetines taikomąsias programas.
- Pritaikyti dinamišką paslaugų formų kūrimą PEPS portale bei sukurti patogią sistemos administratoriaus ir naudotojo sąsają.

Darbo aprobata:

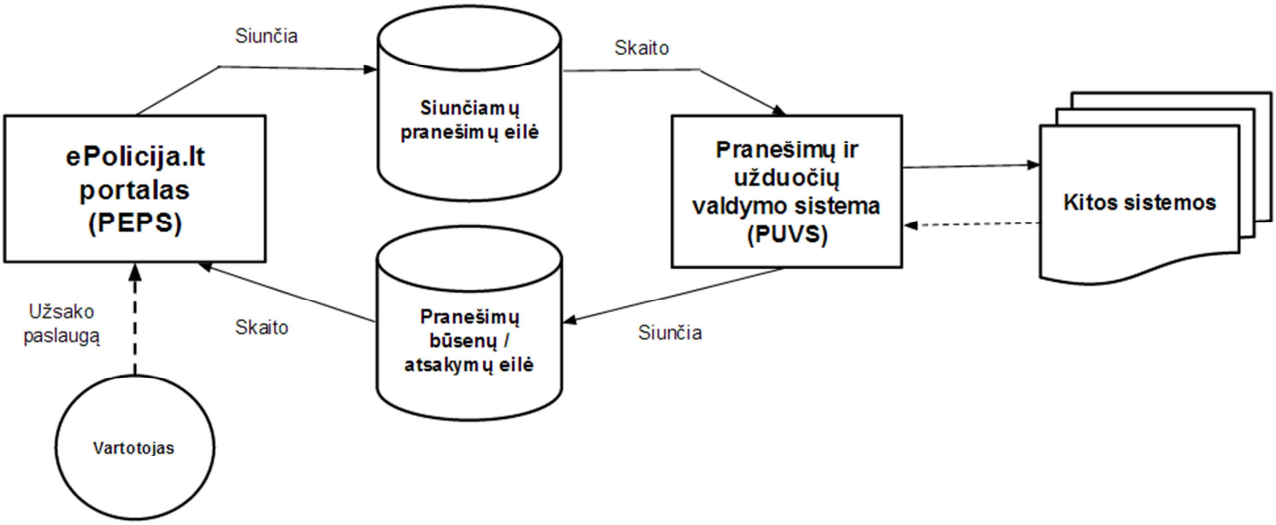
N. Bučinskis. Modernių internetinių aplikacijų kūrimas naudojant Java ir ZK internetinį karkasą. 16-osios Lietuvos jaunųjų mokslininkų teminės konferencijos „Mokslas – Lietuvos ateitis“ straipsnių rinkinys. Vilnius: Technika, 2013 p 6-9 (žr. 1 priedą).

2. EPOLICIJA.LT ANALIZĖ

Policijos elektroninė paslauga yra fiziniam ar juridiniam asmeniui, kuris kreipiasi į policiją, teikiama elektroninė viešoji paslauga, kurios procedūros, skirtos šiai paslaugai gauti, atliekamos pranešėjo buvimo vietoje informacinėmis ir ryšių priemonėmis. Pagrindinė PEPS portalo dalis yra šių paslaugų valdymas. PEPS portalas atlieka tik pirmąjį paslaugos teikimo naudotojams vaidmenį, t.y. yra pradinis sluoksnis, kuriame pateikiamos paslaugų teikimo formos, surenkami duomenys bei išsiunčiami pranešimai tolesniam vykdymui (1 pav.).

Pranešimų valdymu atsakinga policijos pranešimų ir užduočių valdymo sistema (toliau - PUVS). Ji priima siunčiamus pranešimus (pranešimas - iš pranešėjo PEPS portale užpildytos pranešimo formos sukurtas XML ar kitokio formato failas, trumpoji žinutė ar MMS pranešimas) juos analizuoja ir perduoda kitoms sistemoms. Pranešimų bei jų būsenų siuntimas vyksta asinchroniškai, panaudojant JMS eiles. Kiekviena iš sistemų turi išeinančiąsias ir įeinančiąsias eiles, iš kurių skaitomi ir rašomi pranešimai.

Pranešimų valdymo procese išskiriami du pagrindiniai darbuotojai: ikiteisminio tyrimo pareigūnas bei budėtojas. Ikiteisminio tyrimo pareigūnas - asmuo, kuriam įstaigos vadovo įsakymu suteikti įgaliojimai priimti sprendimą pradėti ir (ar) atlikti ikiteisminį tyrimą ir kuris šios įstaigos ar jos padalinio vadovo pavedimu atlieka BPK numatytus ikiteisminio tyrimo veiksmus: siekia nustatyti nusikalstamą veiką padariusį asmenį ir tokios veikos aplinkybes (BPK 18 str.). Budėtojas - pareigūnas, kuriam įstaigos vadovo įsakymu suteikti įgaliojimai priimti į įstaigą atvykusių asmenų žodinius ar rašytinius skundus, pareiškimus ar pranešimus apie nusikalstamą veiką, priimti elektroninio ryšio priemonėmis įstaigoje gautus skundus, pareiškimus ar pranešimus apie nusikalstamą veiką arba informaciją apie galimą nusikalstamą veiką, priimti sprendimą pradėti ikiteisminį tyrimą arba pavesti ikiteisminio tyrimo įstaigos pareigūnui neatidėliotinai patikrinti gautos informacijos apie galimą nusikalstamą veiką tikrumą. Paslaugų valdymo PEPS portale sistemą prižiūri pareigūnas (toliau - administratorius), kuris kuria paslaugų formas, užtikrina naujienų įdėjimą bei naudotojų administravimą.



1 pav. Supaprastinta PEPS pranešimų valdymo schema

PEPS veikimas pagal pateiktą PEPS architektūros schemą (2 pav.):

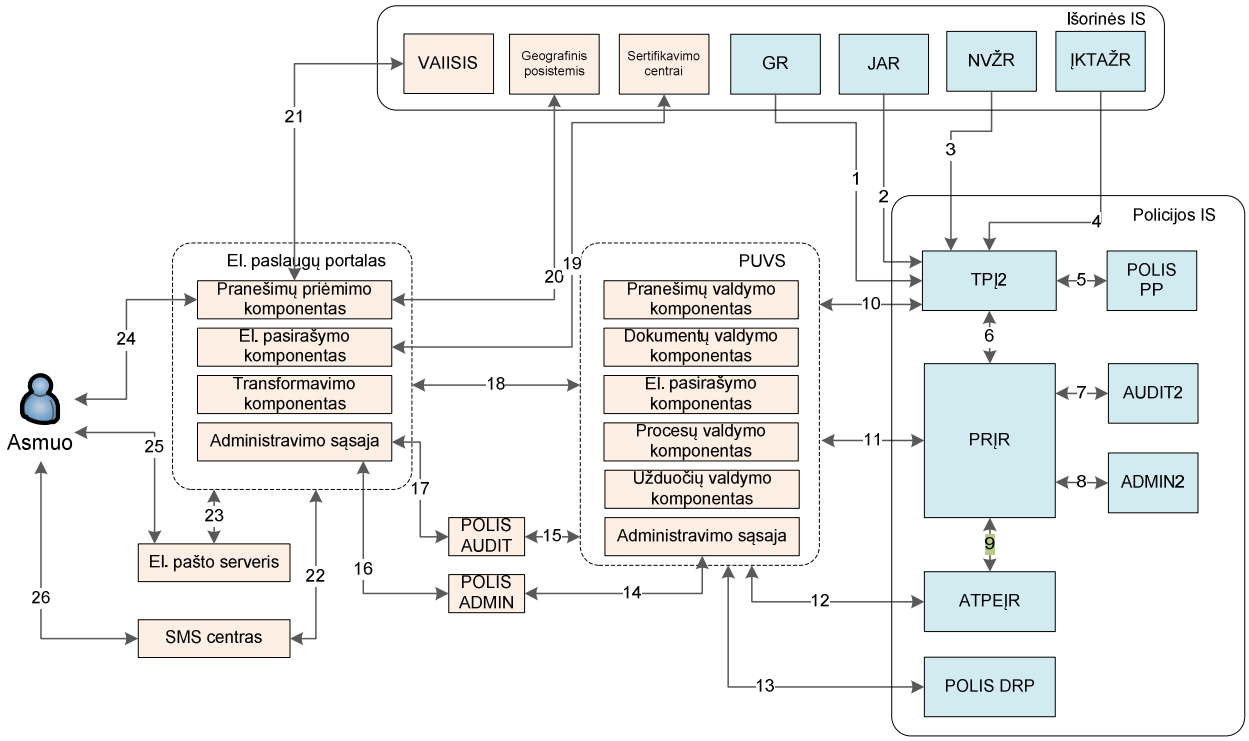
1. Gyventojas identifikuojamas ir nustatoma jo tapatybė VIISP priemonėmis.
2. Po sėkmingo tapatybės nustatymo gyventojas patenka į PEPS portalą.
3. Gyventojas PEPS portale inicijuoja paslaugą, užpildo užsakymo duomenis, jei reikalinga, suformuotą dokumentą pasirašo elektroniniu parašu ir pateikia užsakymą vykdymui.
4. PEPS portalas į PUVS sistemą nusiunčia užsakymo duomenis, el. dokumentą ir susijusią informaciją.
5. PUVS sistemoje sukuriamas naujas paslaugos apdorojimo procesas, kuriame dalyvauja pareigūnai bei išorinės sistemos. Proceso vykdymo metu sukuriamos užduotys atitinkamiems pareigūnams atlikti tam tikrus su paslauga susijusius veiksmus. Taip pat per integralią platformą surenkami arba siunčiami duomenys į kitas susijusias sistemas (PRĮR, ATPEĮR, POLIS DRP ir pan.). Vykstančio proceso jo būsenos perduodamos į PEPS portalą, kad gyventojas matytų savo paslaugos užsakymo eigą.
6. Procesui pasibaigus, į PEPS portalą išsiunčiamas atliktos paslaugos rezultatas: dokumentas, pažyma arba kita susijusi informacija. Gyventojas apie paslaugos įvykdymą gali būti informuojamas telefonu, SMS žinute, el. paštu arba portale.

PEPS pranešimų analizei bei apibendrinimui, naudotojų aktyvumo apskaitai bus naudojamas analizės ir ataskaitų komponentas – priemonė skirta sukauptų duomenų analizei vykdyti pagal nustatytus rodiklius.

PEPS nėra autonominė sistema, ji realizuoja sąsajas su išorinėmis sistemomis bei vidinėmis policijos sistemomis, per kurias įvairiais būdais ir protokolais teikia ir gauna duomenis.

Pagrindiniai procesai siekiamos kompiuterizuojamo objekto būsenos:

- Pranešimo (pareiškimo ar skundo) teikimas - pranešimas teikiamas SMS, MMS žinute, užpildant WEB formą portale ar atsiunčinat el. paštu. Naudotojai identifikuojami ir nustatoma jų tapatybė VIISP priemonėmis.
- Pranešimo tikrinimas bei papildomų duomenų surinkimas - pranešimas peržiūrimas ir duomenys tikslinami remiantis GR, JAR, NVŽR, ĮKTAŽR duomenimis.
- Pranešimo nagrinėjimas - pranešimas nagrinėjamas pareigūno ir registruojamas PRĮR arba POLIS DRP.
- Informavimas pranešimo nagrinėjimo eigą ir rezultatus, apie paslaugos teikimą informuojama SMS žinute, el. paštu arba el. paslaugų portale.



2 pav. PEPS architektūra

2.1. Panaudos atvejai

Norint tinkamai realizuoti sistemą, būtina tiksliai nustatyti panaudos atvejus ir išsiaiškinti jų procesus. Panaudos atvejų analizė užtikrina, jog sistema bus realizuota tinkamai ir nebus praleista veiksmų seka, būtina užbaigtai sistemai.

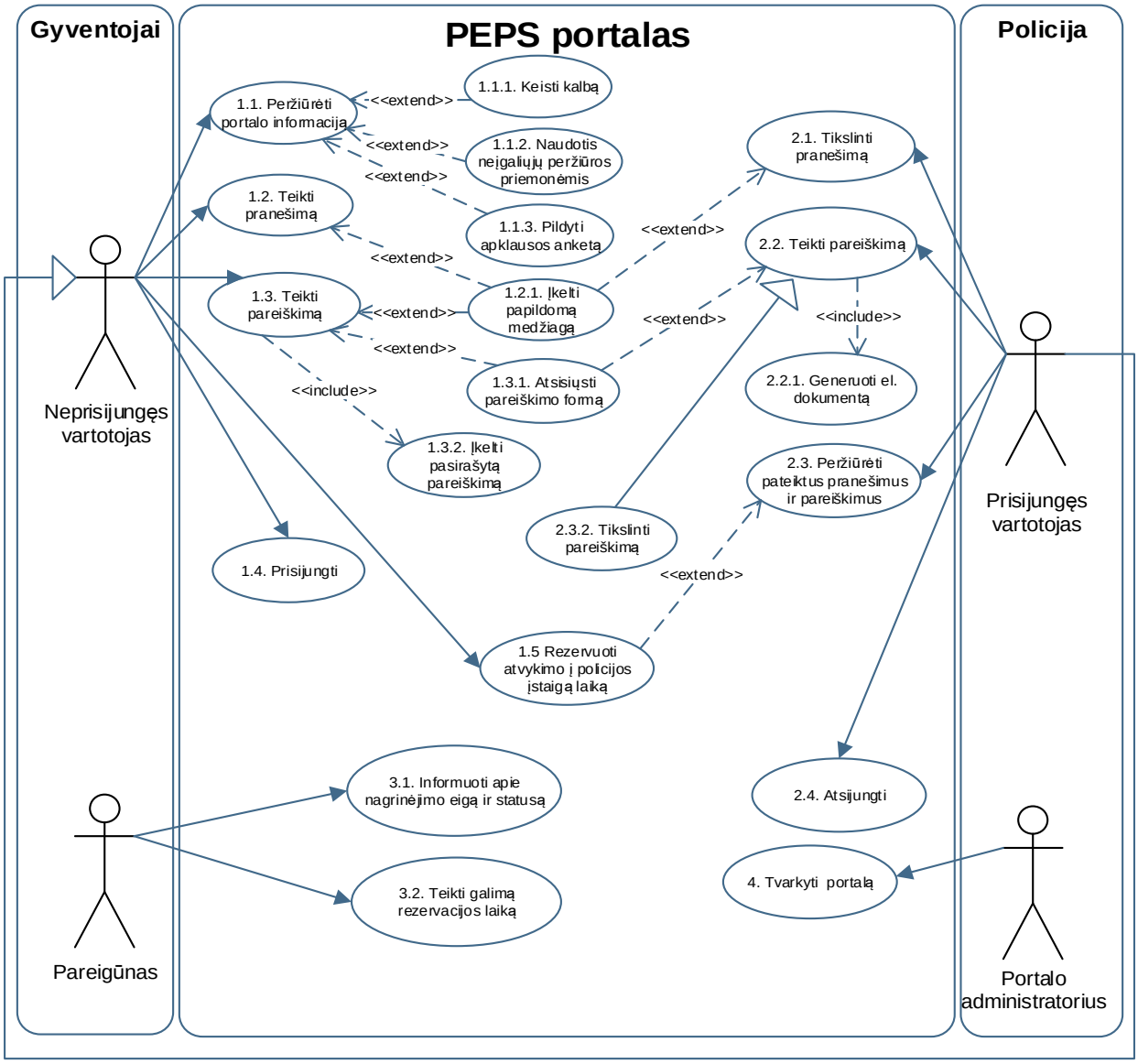
Panaudos atvejuose išskiriami trys pagrindiniai dalyviai: gyventojai arba naudotojai, kurie naudosis e.policijos teikiamomis paslaugomis, policijos darbuotojai, valdantys pranešimų apdorojimą ir pats PEPS portalas (3 pav.). Žemiau pateiktame paveiksle parodyti visi pagrindiniai

veiksmai, kuriuos galima atlikti PEPS portale. Panaudos atvejuose neišskiriamos išorinės sistemos, orientuojamasi į tik naudotojo sąsajos veiksmus.

Tolimesniuose skyriuose detalizuojami šie pagrindiniai panaudos atvejai, susiję su teikiamomis paslaugomis:

- Pranešimo policijai teikimas.
- Pareiškimo, skundo ar prašymo policijai teikimas.

Šie panaudos atvejai yra pagrindiniai duomenų šaltiniai ir viso PEPS portalo pagrindas, kadangi perkelia pagrindines policijos teikiamas paslaugas į elektroninę erdvę. Šiomis paslaugomis paremta visa paslaugų teikimo sistema ir jos yra pirmosios paslaugos su administratoriaus sukurtomis duomenų priėmimo formomis.

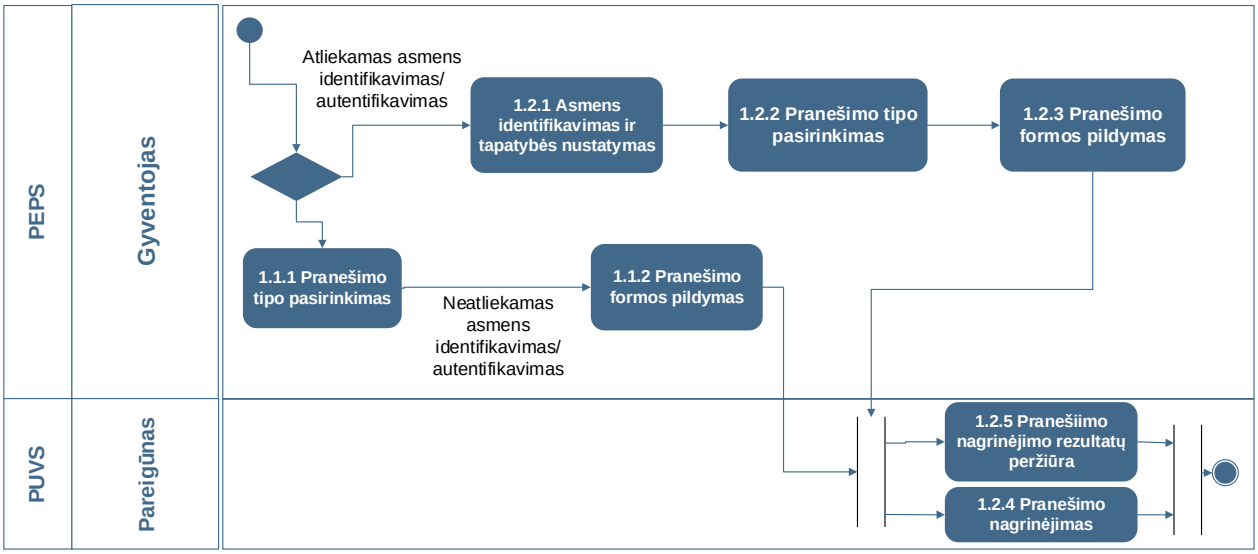


3 pav. Bendra PEPS portalo panaudos atvejų diagrama

1 lentelė. Panaudos atvejai

PA numeris	PA pavadinimas	PA aprašymas
PA1.1.	Peržiūrėti portalo informaciją	Peržiūrima visa vieša portalo informacija.
PA1.1.1.	Keisti kalbą	Keičiama portalo kalba: lietuvių, anglų, rusų.
PA1.1.2.	Naudotis neįgaliųjų svetainės peržiūros priemonėmis	Portalas pritaikomas neįgaliųjų reikmėms.
PA1.1.3.	Pildyti apklausos anketą	Pildoma naudotojų pasitenkinimo anketa, pateikiami apklausos rezultatai. Anketa pildoma naudotojo pasirinkimu arba pateikiama užpildžius pranešimą arba pareiškimo formą.
PA1.2.	Teikti pranešimą	Pranešimo policijai teikimas.
PA1.2.1.	Įkelti papildomą medžiagą	Teikiamas pranešimas ar pareiškimas papildomas multimedijos medžiaga. Apimties ribojimas – 5 MB. 1 vnt., galima pateikti neribotą kiekį rinkmenų.
PA1.3.	Teikti pareiškimą	Pareiškimo, prašymo ar skundo policijai teikimas.
PA1.3.1.	Atsisiųsti pareiškimo formą	Naudotojas gali atsisiųsti teksto redaktoriumi redaguojamą pareiškimo formą, kurią užpildęs ir pasirašęs savo naudojamomis priemonėmis gali prisegti prie pranešimo WEB formos.
PA1.3.2.	Įkelti pasirašytą pareiškimą	Prie WEB formos prisegti savomis priemonėmis sukurtą ir el. parašu pasirašytą el. dokumentą.
PA1.4.	Prisijungti	Prisijungti prie sistemos VIISP teikiamomis asmens tapatybės nustatymo elektroninėje erdvėje priemonėmis.
PA1.5.	Rezervuoti atvykimo į policijos įstaigą laiką	Rezervuoti atvykimo į policijos įstaigą laiką. Naudojama policijos laiko rezervavimo sistema.
PA2.1.	Tikslinti pranešimą	Tikslinti pateiktą pranešimą ir papildyti jį naujais duomenimis. Pranešimo tikslinimo procedūra analogiška naujo pranešimo teikimo procedūrai.
PA2.2.	Teikti pareiškimą	Pareiškimo, prašymo ar skundo policijai teikimas.
PA2.2.1.	Generuoti el. dokumentą	Naudojant pateiktus duomenis generuojamas ADOC V1.0 specifikaciją atitinkantis el. dokumentas, kuris pasirašomas el. parašu.
PA2.3.	Peržiūrėti pateiktus pranešimus ir pareiškimus	Peržiūrėti pateiktų pareiškimų ir pranešimų sąrašą. Pagal pasirinktus parametrus filtruoti pateiktus pranešimus.
PA2.3.2.	Tikslinti pareiškimą	Tikslinti pateiktą pareiškimą ir papildyti jį naujas duomenimis. Pareiškimo tikslinimo procedūra analogiška naujo pareiškimo teikimo procedūrai. Suteikiamas naujas pareiškimo identifikacijos numeris, kuris susiejamas su tikslinamo pareiškimo numeriu.
PA2.4.	Atsijungti	Atsijungti nuo sistemos.
PA3.1.	Informuoti apie nagrinėjimo eigą ir statusą	Informuoti apie pranešimo eigą galima šiais būdais: el. paštu, SMS žinute, portale.
PA3.2.	Teikti galimą rezervacijos laiką	Pildyti galimą gyventojų priėjimo laiką.
PA4.	Tvarkyti portalą	Keisti portalo nustatymus, kurti paslaugų formas.

2.1.1. Pranešimo policijai teikimas

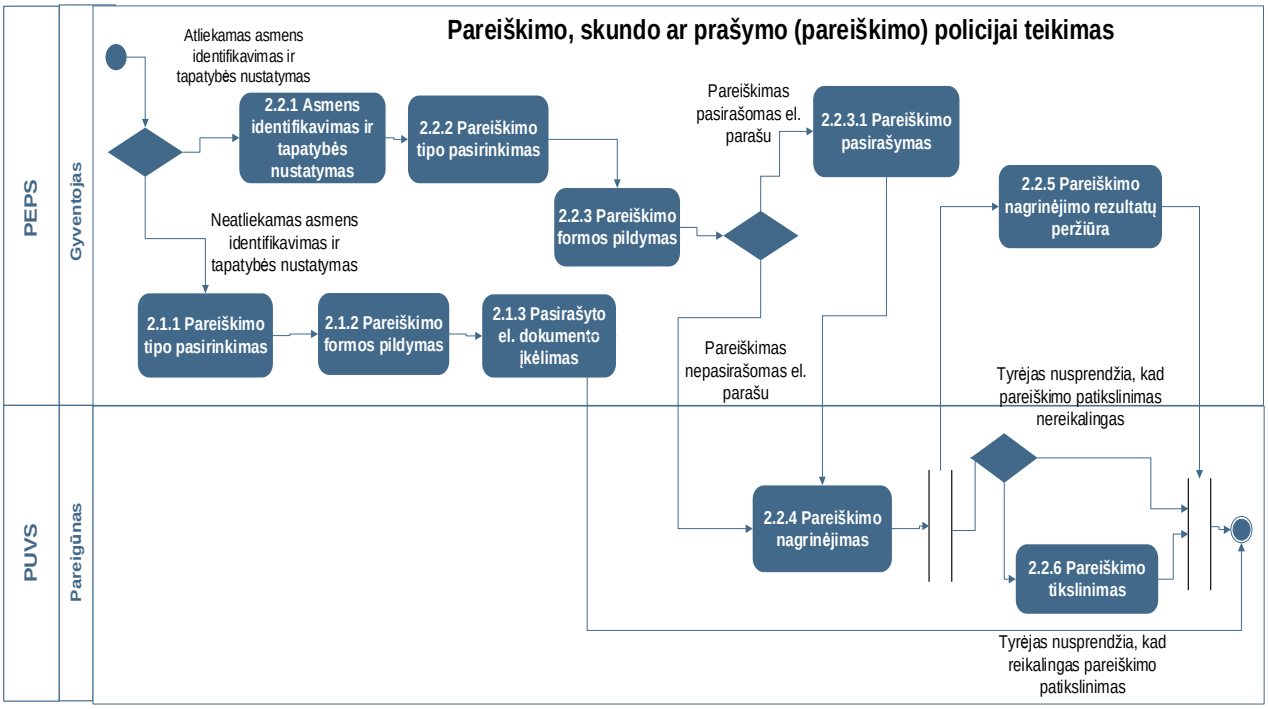


4 pav. Pranešimo teikimo policijai diagrama

2 lentelė. Pranešimo teikimo policijai aprašymas

Numeris	Pavadinimas	Aprašymas	Perėjimo sąlygos
DS1.1.1.	Pranešimo tipo pasirinkimas	Pranešimo tipas pasirenkamas iš baigtinio sąrašo.	-
DS1.1.2.	Pranešimo formos pildymas	Pildoma struktūrizuota pranešimo forma	Užpildyti visi privalomi pranešimo formos langai
DS1.2.1.	Asmens identifikavimas ir tapatybės nustatymas	Asmuo identifikuojamas VIISP asmens tapatybės nustatymo elektroninėje erdvėje priemonėmis	Identifikavimas sėkmingas. VIISP persiunčia naudotojo duomenis (a.k.)
DS1.2.2.	Pranešimo tipo pasirinkimas	Pranešimo tipas pasirenkamas iš baigtinio sąrašo.	-
DS1.2.3.	Pranešimo formos pildymas	Pildoma struktūrizuota pranešimo forma. Forma automatiškai užpildoma identifikavimo metu gautais naudotojo duomenimis ir naudotojo sukurtais paskyros duomenimis (jei duomenys pateikti)	Užpildyti visi privalomi pranešimo formos langai
DS1.2.4.	Pranešimo nagrinėjimas	Nagrinėjamas pranešimas, su nagrinėjimu susijusi informacija pateikiama portale	-
DS1.2.5.	Pranešimo nagrinėjimo rezultatų peržiūra	Peržiūrima informacija apie pranešimo nagrinėjimą	Pranešimui suteikiamas identifikacinis numeris

2.1.2. Pareiškimo, skundo ar prašymo policijai teikimas



5 pav. Pareiškimo ar skundo teikimo diagrama

3 lentelė. Pareiškimo ar skundo teikimo aprašymas

Numeris	Pavadinimas	Aprašymas	Perėjimo sąlygos
DS2.1.1.	Pareiškimo tipo pasirinkimas	Pareiškimo tipas pasirenkamas iš baigtinio sąrašo	-
DS2.1.2.	Pareiškimo formos pildymas	Pildoma struktūrizuota pareiškimo forma	Užpildyti visi privalomi pareiškimo formos langai
DS2.1.3.	Pasirašyto el. dokumento įkėlimas	Prie pareiškimo formos prisegamas asmens naudojamomis priemonėmis sukurtas ADOC V1.0. specifikaciją atitinkantis el. dokumentas. Leidžiama prisegti tik 1 el. dokumentą	El. dokumento įkėlimas sėkmingas.
DS2.2.1.	Asmens identifikavimas ir tapatybės nustatymas	Asmuo identifikuojamas VIISP asmens tapatybės nustatymo elektroninėje erdvėje priemonėmis	Identifikavimas sėkmingas. VIISP persiunčia naudotojo duomenis (a.k.)
DS2.2.2.	Pareiškimo tipo pasirinkimas	Pareiškimo tipas pasirenkamas iš baigtinio sąrašo	-
DS2.2.3.	Pareiškimo formos pildymas	Pildoma struktūrizuota pareiškimo forma. Forma automatiškai užpildoma identifikavimo metu gautais naudotojo duomenimis ir naudotojo sukurtais paskyros duomenimis (jei duomenys pateikti)	Užpildyti visi privalomi pareiškimo formos langai

3 lentelės tęsinys

Numeris	Pavadinimas	Aprašymas	Perėjimo sąlygos
DS2.2.3.1.	Pareiškimo pasirašymas	Iš užpildytos WEB formos ir kitų pateiktų duomenų formuojamas ADOC V1.0 specifikaciją atitinkantis el. dokumentas, kuris pasirašomas elektroniniu parašu	El. dokumento generavimas ir pasirašymas sėkmingi.
DS2.2.4.	Pareiškimo nagrinėjimas	Nagrinėjamas pareiškimas, su nagrinėjimu susijusi informacija pateikiama portale	Pareiškimui suteikiamas identifikacinis numeris
DS2.2.5.	Pranešimo nagrinėjimo rezultatų peržiūra	Peržiūrima informacija apie pranešimo nagrinėjimą	Užpildomi visi privalomi pareiškimo formos langai
DS2.2.6.	Pareiškimo tikslinimas	Tikslinti pateiktą pareiškimą ir papildyti jį naujas duomenimis. Pareiškimo tikslinimo procedūra analogiška naujo pareiškimo teikimo procedūrai. Suteikiamas naujas pareiškimo identifikacinis numeris, kuris susiejamas su tikslinamo pareiškimo numeriu	

2.2. Asmenų identifikavimas ir tapatybės nustatymas

Asmenų identifikavimas užtikrinamas naudojant VIISP teikiamą tapatybės nustatymo paslaugą. Naudotojai identifikuojami remiantis viešojo administravimo institucijų sistemos funkcionavimo taisyklėmis, patvirtintomis Informacinės visuomenės plėtros komiteto prie Lietuvos Respublikos Vyriausybės direktoriaus 2008-12-18 įsakymu Nr. T-228.

Asmenų identifikavimui naudojamos visos galimos priemonės (skaitmeniniai gyventojų ir valstybės tarnautojų sertifikatai, elektroninės bankininkystės sistemos). Asmenų identifikavimas atliekamas tiek jungiantis iš PEPS elektroninių paslaugų portalo.

2.3. El. dokumento kūrimas ir pasirašymas

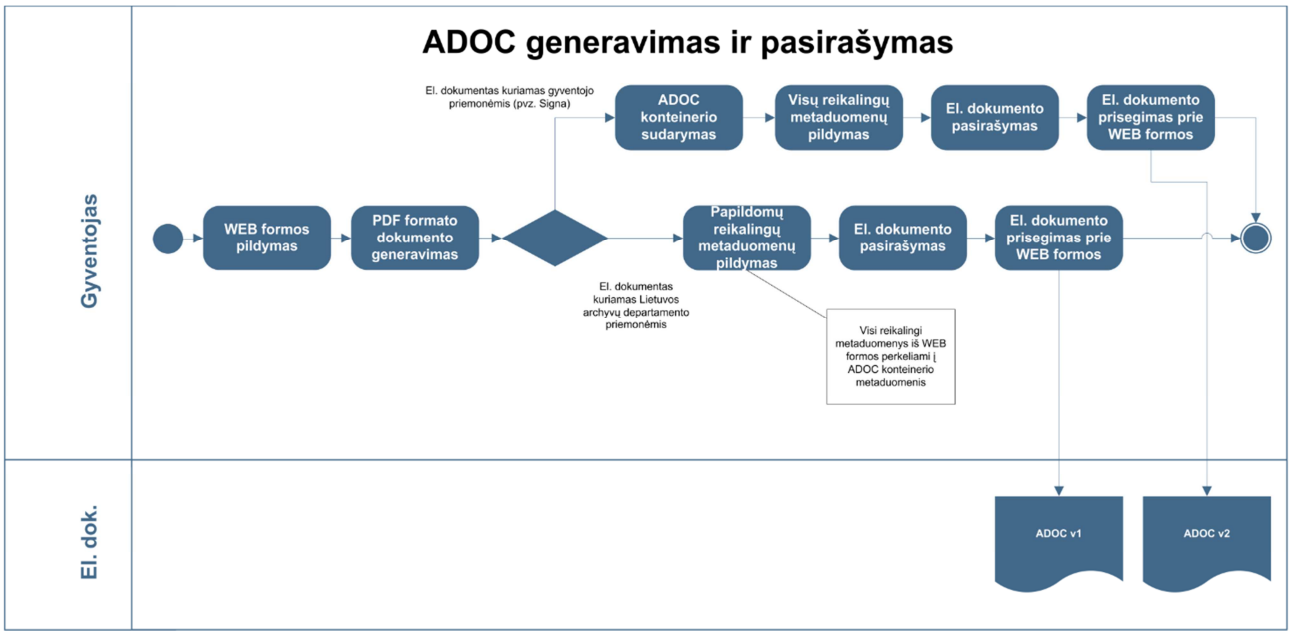
El. dokumentas kuriamas ir pasirašomas išorinėmis programinėmis priemonėmis, tam galimi du el. dokumento kūrimo ir pasirašymo būdai (6 pav.):

- asmens pasirinktos ir dažniausiai naudojamos priemonės (SIGNA, e. paslaugų portalas).
- Lietuvos vyriausiojo archyvaro tarnybos laisvai platinamos programinės priemonės.

Abiem atvejais iš užpildytos WEB formos yra suformuojamas *.pdf* formato dokumentas, kuris naudojamas kaip pagrindinis ADOC konteinerio dokumentas. Siekiant užtikrinti WEB formos ir elektroninio dokumento atitikimą *.pdf* formato dokumento turinyje nurodomas pranešimo ar pareiškimo OID (unikalus ID).

Pirmu atveju asmuo *.pdf* formato el. dokumentą parsisiunčia į savo darbo vietos kompiuterį, jį naudoja kaip pagrindinį ADOC konteinerio dokumentą, užpildo privalomus metaduomenis bei pasirašo el. dokumentą. Suformuotas ADOC formato el. dokumentas prisegamas prie užpildytos WEB formos.

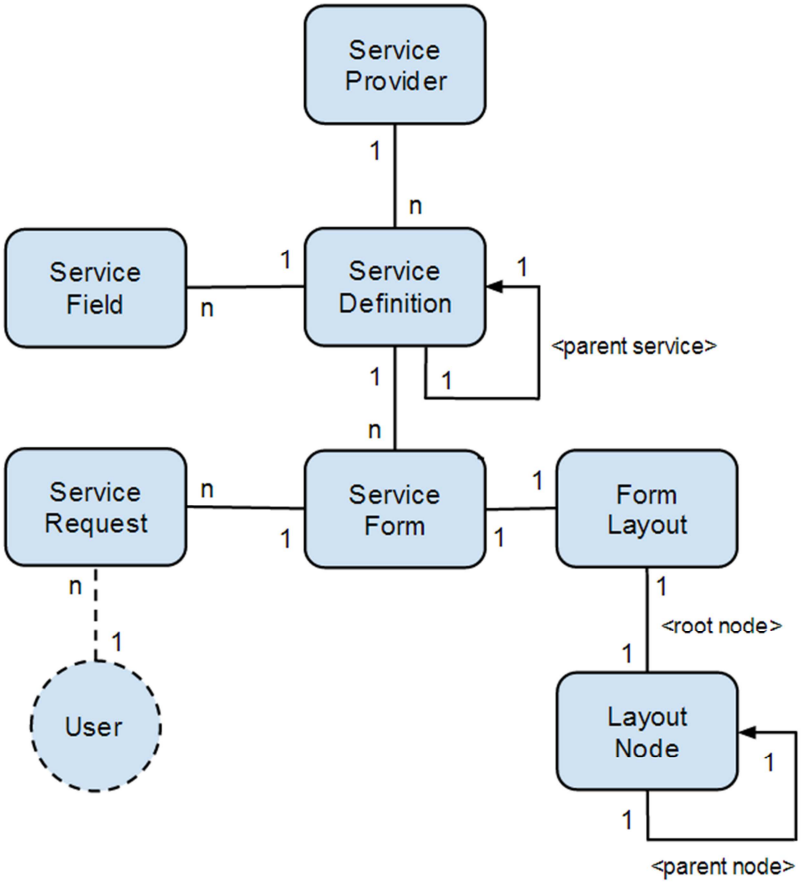
Antruoju atveju naudojamos Lietuvos vyriausiojo archyvaro tarnybos programinės priemonės. Viena iš programinių priemonių galimybių naudoti el. parašą yra naudotojų iš trečiųjų šalių nukreipimas į elektroninio archyvo sistemą, tinklinėmis paslaugomis perduodant metaduomenis ir el. dokumento turinį. Šiuo atveju, *.pdf* formato dokumentas bei dalis metaduomenų būtų perduodama iš PEPS tinklinėmis paslaugomis. Suformuotą ir pasirašytą el. dokumentą asmuo prisegtų prie užpildytos WEB formos.



6 pav. ADOC generavimo ir pasirašymo diagrama

2.4. Duomenų bazės modelis

Paslaugų modelį PEPS portale sudaro 7 pagrindinės duomenų bazės lentelės, kurios išlaiko visą tiekiamų paslaugų struktūrą. Pagrindinė lentelė, kurioje saugomi paslaugų užsakymų įrašai yra „SERVICE_REQUEST“. Lentelės „LAYOUT“ ir „LAYOUT_NODE“, kurios išsaugo formų struktūrą, visiškai atskirtos nuo paslaugų, todėl gali būti pritaikomos kitose sistemose. Konceptualinė esybių schema parodyta žemiau (**7** pav.).



7 pav. Esybių (lentelių) sąryšio schema

2.4.1. Lentelė „SERVICE_REQUEST“

Ši lentelė yra pagrindinė, kurioje saugomi portale surinkti duomenys apie paslaugos užsakymą. Į ją taip pat rašomi įrašai paslaugas užsakant kitais būdais, t.y. per SMS, MMS ar el. paštą. Laukelyje „FORM_DATA“ įrašoma į XML struktūrą paverstas rakto ir reikšmės tipo objektas (angl. *Map*), kuriame saugomos rakto ir reikšmės poros. Reikšmės pagal raktą atitinka konkretų komponentą formoje.

Žemiau pateikiamas visų lentelės laukų aprašymas (4 lentelė).

4 lentelė. Duomenų bazės lentelės „SERVICE_REQUEST“ laukai.

Stulpelio pavadinimas	Privaloma	Aprašymas
ID	Taip	Vidinis įrašo ID
CREATE_DATE	Taip	Įrašo sukūrimo data (atitinka paslaugos užsakymo datą)
EDIT_DATE	Ne	Įrašo keitimo data
STATE	Taip	Paslaugos užsakymo būsena. Galimos reikšmės: • REQUEST_ACCEPTED • REQUEST_REJECTED • SERVICE_REQUESTED • SERVICE_IN_PROGRESS • SERVICE_FINISHED

4 lentelės tęsinys

Stulpelio pavadinimas	Privaloma	Aprašymas
USER_ID	Ne	Naudotojo, su kuriuo susietas šis paslaugos užsakymas, ID. Gali nebūti, jei paslauga užsakoma anonimiškai.
UUID	Taip	Unikalus ID, rodomas išorinėse sistemose
VERSION	Ne	Vidinė įrašo versija
SERVICE_FORM_ID	Ne	Paslaugos formos ID, su kuria buvo pateiktas paslaugos užsakymas
FORM_DATA	Ne	Formoje pateikti duomenys.
IP_ADDRESS	Ne	IP adresas, kuris nustatytas paslaugos užsakymo metu.
SENDER_TYPE	Taip	Siuntėjo tipas: • PORTAL • EMAIL • SMS • MMS
FORM_DATA_ADOC	Ne	Pateikti duomenys, kurie pasirašyti elektroniniu parašu.
ADOC_FILE_ID	Ne	Nuoroda į el. parašu pasirašytą dokumentą.
SERVICE_DEFINITION_ID	Taip	Nuoroda į paslaugos aprašymą.

2.4.2. Lentelė „SERVICE_DEFINITION“

Lentelė, sauganti paslaugos informaciją. Joje saugomi duomenys apie paslaugos galiojimą, statusą, kokia yra einamoji forma ir kt. (5 lentelė).

Objektas, kuris saugomas šioje lentelėje, neaprašo pačios paslaugos logikos. Visos procedūros, kurios atliekamos ją užsakius, identifikuojamos išorinėje sistemoje pagal unikalų kodą, priskiriamą šiai paslaugai.

5 lentelė. Duomenų bazės lentelės „SERVICE_DEFINITION“ laukai

Stulpelio pavadinimas	Privaloma	Aprašymas
ID	Taip	Vidinis įrašo ID
ACCESS_TYPE	Taip	Paslaugos pasiekiamumo lygis: • ANONYMOUS • AUTHENTICATED
CURRENT_FORM_VERSION	Ne	Einamosios duomenų priėmimo formos versija
DESCRIPTION	Ne	Paslaugos aprašymas
NAME	Ne	Paslaugos pavadinimas
STATUS	Taip	Paslaugos statusas: • ACTIVE • INACTIVE
VALID_FROM	Ne	Paslaugos galiojimo pradžia
VALID_TILL	Ne	Paslaugos galiojimo pabaiga
VERSION	Ne	Vidinė įrašo versija
SERVICE_PROVIDER_ID	Taip	Nuoroda į paslaugos tiekėjo objektą
CURRENT_FORM_ID	Ne	Nuoroda į einamąją paslaugos formą

5 lentelės tęsinys

Stulpelio pavadinimas	Privaloma	Aprašymas
CODE	Taip	Unikalus paslaugos kodas, naudojamas identifikuoti paslaugos užsakymams išorinėse sistemose
VISIBILITY_TYPE	Taip	Paslaugos matomumo lygis: • PUBLIC • PRIVATE
NAME_RESOURCE_ID	Ne	Nuoroda į lokalizuotą paslaugos pavadinimą (naudojamas parodyti paslaugos pavadinimą vartotojui pagal pasirinktą kalbą)
PARENT_SERVICE_ID	Ne	Nuoroda į tėvinę paslaugą

2.4.3. Lentelė „SERVICE_PROVIDER“

Informacija apie paslaugos tiekėją. Šis lentelė aprašo objektą, kuris atlieka rūšiavimo vaidmenį, t.y. suskirtos paslaugas į tam tikras grupes (6 lentelė).

6 lentelė. Duomenų bazės lentelės „SERVICE_PROVIDER“ laukai

Stulpelio pavadinimas	Privaloma	Aprašymas
ID	Taip	Vidinis įrašo ID
DESCRIPTION	Ne	Paslaugos tiekėjo aprašymas
NAME	Ne	Paslaugos tiekėjo pavadinimas
VERSION	Ne	Vidinė įrašo versija
CODE	Taip	Paslaugos tiekėjo kodas

2.4.4. Lentelė „SERVICE_FIELD“

Lentelė saugoti paslaugų formų laukų šablonams. Šablonas susideda iš trijų pagrindinių dalių: modelio rakto (su kokiu raktu susieta reikšmė modelyje), pavadinimo ir tipo (7 lentelė). Šablonai atlieka naudojamų laukų tipų vientisumą, kuomet jie naudojami daugelį kartų skirtingose formose.

7 lentelė. Duomenų bazės lentelės „SERVICE_FIELD“ laukai

Stulpelio pavadinimas	Privaloma	Aprašymas
ID	Taip	Vidinis įrašo ID
DESCRIPTION	Ne	Formos lauko šablono aprašymas
MODEL_PATH	Taip	Šablono modelio raktas
NAME	Taip	Formos lauko šablono pavadinimas
VERSION	Ne	Vidinė įrašo versija
SERVICE_DEFINITION_ID	Taip	Nuoroda į paslaugos aprašymą
VALUE_DATA_TYPE_ID	Taip	Nuoroda į lauko tipo objektą

2.4.5. Lentelė „SERVICE_FORM“

Lentelė, kurioje saugomos paslaugų formos. Joje talpinama informacija apie formą, tokia kaip formos versija, pavadinimas ir kt. Formos struktūra gaunama iš nuorodos į „FORM_LAYOUT“ objektą (8 lentelė).

8 lentelė. Duomenų bazės lentelės „SERVICE_FORM“ laukai

Stulpelio pavadinimas	Privaloma	Aprašymas
ID	Taip	Vidinis įrašo ID
CREATE_DATE	Taip	Formos sukūrimo data
DESCRIPTION	Ne	Formos aprašymas
EDIT_DATE	Taip	Formos koregavimo data
FORM_VERSION	Taip	Formos versija
NAME	Taip	Formos pavadinimas
VERSION	Ne	Vidinė įrašo versija
FORM_LAYOUT_ID	Taip	Nuoroda į formos struktūrą
SERVICE_DEFINITION_ID	Taip	Nuoroda į paslaugos aprašymą
CAPTCHA_PROTECTED	Ne	Savybė, nurodanti ar reikia apsaugoti duomenis nuo automatizuoto įvedimo
SIGNATURE_TYPE	Taip	Pasirašymo tipas: • NONE • OPTIONAL_EXPANDED • OPTIONAL_COLLAPSED
WIZARD_MODE	Ne	Savybė, nurodanti ar formą rodyti vedlio režimu

2.4.6. Lentelė „FORM_LAYOUT“

Paslaugos formos struktūrą sauganti lentelė. Struktūra yra nustatoma pagal individualaus elemento tėvinį elementą, todėl šios lentelės įraše saugomas tik aukščiausiai esančio elemento ID (9 lentelė).

9 lentelė. Duomenų bazės lentelės „FORM_LAYOUT“ laukai

Stulpelio pavadinimas	Privaloma	Aprašymas
ID	Taip	Vidinis įrašo ID
LAST_CHANGE_DATE	Ne	Paskutinio keitimo data
VERSION	Ne	Vidinė įrašo versija
ROOT_NODE_ID	Taip	Nuoroda į aukščiausiai struktūroje esantį elementą

2.4.7. Lentelė „LAYOUT_NODE“

Lentelė, kurioje saugomi visi paslaugos formos elementai. Kiekvienas elementas siejasi su kitu per „PARENT_ELEMENT_ID“, kuriame yra nuoroda į tėvinį elementą. Visa struktūra nustatoma sekančiu algoritmu: paimamas komponentas su ID pagal „FORM_LAYOUT“ lentelės „ROOT_NODE_ID“, ir jam pridedami visi elementai kurie turi tokį ID savo „PARENT_ELEMENT_ID“ laukelyje. Tas pats kartojama visiems išrinktiems komponentams kol neberandamas kitas komponentas kuris turėtų „PARENT_ELEMENT_ID“ pagal aukštesnio komponento ID.

Visi komponento laukai nurodyti žemiau (10 lentelė).

10 lentelė. Duomenų bazės lentelės „LAYOUT_NODE“ laukai

Stulpelio pavadinimas	Privaloma	Aprašymas
ID	Taip	Vidinis įrašo ID
DEFAULT_VALUE	Ne	Reikšmė pagal nutylėjimą
LABEL_LAYOUT_TYPE	Taip	Komponento etiketės tipas: • VERTICAL • HORIZONTAL • NONE
MAX_VALUE	Ne	Maksimali leidžiama reikšmė
MIN_VALUE	Ne	Minimali leidžiama reikšmė
MODEL_PATH	Ne	Reikšmės raktas modelyje
REQUIRED	Ne	Ar reikšmę privaloma įvesti
CONVERSION_PATTERN	Ne	Modelio reikšmės keitimo šablonas
VERSION	Ne	Vidinė įrašo versija
MODEL_DATA_TYPE_ID	Ne	Nuoroda į modelio reikšmės tipą
PARENT_ELEMENT_ID	Ne	Nuoroda į tėvinį komponentą
COMPONENT_TEMPLATE_ID	Taip	Nuoroda į komponento šabloną
NODE_INDEX	Ne	Komponento indeksas

2.5. Technologijų analizė

Prieš pradedant analizuoti technologijas, būtina išrinkti tinkamiausią programavimo kalbą. Internetines aplikacija galima kurti su bet kuria programavimo kalba, skiriasi tik sudėtingumo lygis. PEPS portalui, pagal kuriančiosios įmonės patirtį, buvo parinkta Java programavimo kalba.

Java, kuriama ir prižiūrima „Oracle Corporation“, 2012 metų duomenimis buvo pati populiariausia programavimo kalba pasaulyje. Pastaruoju metu Java siejama su Android mobiliąja platforma, tačiau ji geriausiai žinoma ir naudojama kliento-serverio internetinėms aplikacijoms kurti. Renkantis Java pagrindu paremtas technologijas, gaunami šie privalumai:

- atviro kodo bibliotekos. Java turi plačią programuotojų bendruomenę, 2012 metais siekusią 10 mln. naudotojų, kuri dalijasi žiniomis, patarimais bei laisvai naudojamais kodo moduliais;
- platus internetinių karkasų pasirinkimas. Jų naudojimas leidžia supaprastinti aplikacijos kūrimą, ypač susijusį su grafine naudotojo sąsaja;
- greitas ir efektyvus produkto kūrimo ciklas. Dėl paprastos kodo sintaksės ir daugybės pasiekiamų pagalbinių modulių, produktas greitai pasiekia diegimo į testavimo aplinkos etapą [14].

Sukurti funkcionalią internetinę sistemą, leidžiančią patogiai valdyti turinį, paslaugas bei pačią aplikaciją, reikia išrinkti tinkamiausias technologijas iš šių grupių:

- Aplikacijos talpinimo sistema (aplikacijos aplinka). Aplikacijos talpinimo sistema tai serveris, arba serverio programėlių (angl. *Servlet*) konteineris, kurioje diegiama sukurta internetinė aplikacija. Tai gali būti paprastas ir populiarus „Apache Tomcat“ serveris, arba

sudėtingesnis ir didesnis, pavyzdžiui „Liferay“ arba „IBM WebSphere“ aplikacijų portalas. Paprasti serveriai suteikia pagrindinį funkcionalumą, reikalingą aplikacijai priimti HTTP užklausas ir jas padoroti, tačiau renkantis tokį sprendimą, reikalinga sukurti daugiau naudotojo sąsajos, ypač jei naudojamas turinio ar naudotojų valdymas. Tačiau toks funkcionalumas jau būna realizuotas moderniuose portaluose, todėl naudojant juos nereikia skirti papildomų pastangų ir laiko.

- Aplikacijos grafinės sąsajos karkasas. Kuriant internetinę aplikaciją neišsiverčiama be internetinio karkaso. Internetinis karkasas suteikia pagrindą, kuriuo naudojantis nereikia kurti žemo lygio funkcijų, tokių kaip reikšmių paėmimas iš puslapio laukų, puslapio dalių atnaujinimas ir t.t. Daugelis internetinių karkasų papildo HTML kodo rašymą papildomomis galimybėmis, tokie kaip „Apache Wicket“ ar „Grails“, tačiau yra tokių, kuriais remiasi tik XML struktūromis, pavyzdžiui „ZK“.
- Aplikacijos vidinės (verslo) logikos bibliotekos bei moduliai.

2.5.1. Portalo pagrindas

Portalas – internetinis aplikacijų serveris, palaikantis portalo įskiepių specifikacijas. Jis veikia kaip portalo įskiepių konteineris, talpindamas portalo įskiepius, kontroliuodamas jų išdėstymą bei valdydamas tarpusavio sąsają. Elektroninių paslaugų portalą epolicija.lt kurti portalo įskiepių konteinerio pagrindu, buvo nuspręsta dėl šių priežasčių:

- Reikalingas lengvas turinio valdymas. Kadangi tai viešai pasiekiamas puslapis, reikalingas reguliarus naujienų talpinimas, atnaujinimas ir šalinimas.
- Teikiamų paslaugų skaičius dažnai kintantis, todėl reikalinga galimybė, nenutraukiant viso portalo veiklos pridėti naujas paslaugas. Tą galima pasiekti sukuriant naują puslapį ir į jį pridedant portalo įskiepį su naujos paslaugos forma.

Portalo įskiepiai (angl. *Portlet*) – atskiri programiniai naudotojo sąsajos elementai, kuriuos valdo ir atvaizduoja internetinis portalas. PEPS naudojami Java portalo įskiepių specifikacijas JSR 168 ir JSR 286 atitinkantys portalo įskiepiai. Šios specifikacijos apibrėžia bendrą aplikacijų programavimo sąsają (angl. *Application Programming Interface, API*), nurodančią, kaip portalo įskiepiai turi bendrauti su portalo įskiepių konteineriu ir įgalina portalo įskiepių tarpusavio bendravimą [7].

Portalo įskiepių pagrindu grindžiamas paslaugos formos talpinimas į konkretų portalo puslapį. Kiekvienas portalo įskiepis palaiko konfigūraciją, todėl patogu nustatyti parametrus, nenaudojant papildomos sąsajos.

Rinkoje internetinių portalų yra gana daug. Išrinkus tik Java EE pagrindu veikiančius portalus, gaunama porą dešimčių skirtingų portalų, besiskiriančių standartinių portalo įskiepių gausa ir funkcionalumu (11 lentelė). Didžioji dalis siūlo mokamas licenzija, tačiau yra keletas veikiančių pagal atviro kodo licenzijas ar LGPL (angl. *Lesser General Public License*).

PEPS portalo pagrindą kurti buvo pasirinktas „Liferay“ portalas, kadangi jis palaiko JSR-168 ir JSR-286 portalo įskiepių specifikacijas bei siūlo saugų ir patikimą sprendimą net ir su nemokama licenzija.

11 lentelė. Internetinių portalų palyginimas

Leidėjas	Pavadinimas	Programavimo kalba	Licenzija	Portalo įskiepių standartai
Apache Software Foundation	Jetspeed 2.2.0	Java EE	Apache License v2.0	JSR-286
ATG	ATG Portal	Java EE	Mokama	JSR-168
Broadvision	Broadvision Portal 8.2	Java EE	Mokama	JSR-168
Bluenog	Bluenog ICE 4.5	Java EE	Mokama	JSR-168
eXo	eXo Portal 2.5	Java EE	Affero General Public License	JSR-286
eXo	eXo Platform 3	Java EE	Mokama	JSR-286
JBoss ir eXo	GateIn Portal 3	Java EE	LGPL	JSR-286
IBM	WebSphere Portal 8.0	Java EE	Mokama	JSR-286, WSRP
Interwoven	TeamPortal	Java EE	Mokama	JSR-168
JBoss	JBoss Enterprise Portal Platform 5.1	Java EE	LGPL	JSR-286, WSRP
Liferay	Liferay Portal 6.1	Java EE	LGPL ir mokamos licenzijos	JSR-168, JSR-286
Open Text Corporation	Vignette Portal 8.0	Java EE	Mokama	JSR-286
Oracle	Oracle WebCenter Suite 11g	Java EE	Mokama	JSR-168, JSR-286, WSRP
Oracle (BEA Systems)	Oracle WebLogic Portal 10g	Java EE	Mokama	JSR-168, JSR-286
Oracle	Oracle IAS Portal 10g	Java EE	Mokama	JSR-168
Samsung SDS	ACUBE Portal 5.0	Java EE	Mokama	JSR-168
SAP AG	SAP NetWeaver 7.0	Java EE	Mokama	JSR-168
Tibco Software	PortalBuilder 5.2	Java EE	Mokama	JSR-168
TmaxSoft	ProPortal 4.0	Java EE	Mokama	JSR-168

Naudojant internetinį portalą, o ne kuriant reikiamą funkcionalumą patiems, atsiranda suderinamumo problemos, jei didelė dalis sistemos yra kuriama JavaScript paremtu karkasu. Dėl savo sudėtingumo, dideli portalo įskiepiai dažnai kertasi su standartiniu portalo funkcionalumu ir

atsiranda klaidos, kurias tenka aiškintis ir taisyti. Tai tampa labai sudėtinga kai reikia nagrinėti didžiulį kiekį išeities kodo, kas ne retai būna dideliuose portaluose.

2.5.1.1. „Liferay“

Vienas iš populiariausių Java pagrindu sukurtų portalų yra „Liferay“. Be standartinio funkcionalumo, kūrėjai dar suteikia didelį bendradarbiavimo priemonių (angl. *Collaboration tools*) rinkinį, specialiai „Liferay“ modifikavimui skirtą programavimo aplinką, įvairių duomenų bazių, programų serverių (angl. *Application servers*) ir operacinių sistemų palaikymą.

Kūrėjai vartotojams siūlo dvi portalo versijas: nemokamą (su GNU LGPL licenzija) „Community edition“ ir mokamą „Enterprise edition“ su klientų aptarnavimu. Šiam projektui naudoti buvo pasirinkta nemokama versija. Pagrindinės priežastys, paskatinusios pasirinkti „Liferay“, o ne kurį kitą konkuruojantį produktą, buvo „Liferay“ siūloma integruota turinio valdymo sistema su įvairių specializuotų turinio tipų palaikymu ir didelis kiekis standartinių portalo įskiepių, kurie arba visiškai atitinka kliento poreikius, arba gali būti atitinkamai modifikuoti [9].

Kartu su „Liferay“ platinamas įskiepių SDK (angl. *Software Development Kit*) – galingą „Liferay“ tobulinimui skirtą įrankį, skirtą programinių modulių kūrimui. Jis leidžia kurti įvairių rūšių priedus (angl. *Plugins*), skirtus išplėsti ar pakeisti „Liferay“ funkcionalumą ir išvaizdą. Galimi priedų tipai:

- tema (angl. *Theme*) – visam portalui suteikimas stilių rinkinys (HTML, CSS, paveiksliukai). Tema taip pat parūpina navigaciją ir leidžia pateikti statinį turinį ar statiškai į tinkamą vietą įterptus portalo įskiepius.
- portalo įskiepis – pagrindinės portalo sudedamosios dalys, suteikiančios didžiąją dalį funkcionalumo. Juos galima kurti bet kuria programavimo kalba, tereikia atitikti JSR 168 ir JSR 286 specifikaciją.
- kablys (angl. *Hook*) – leidžia nekeičiant pačio portalo išeities kodo, pakeisti jo funkcionalumą. Palaiko naujų portalo kalbų, nustatymų, JSP failų sukūrimą ar pakeitimą kitais. Taip pat įgalina programuotoją pakeisti „Struts“ karkaso failus, nurodant vietoje standartinių servisų klasių naudoti naudotojo apibrėžtas klases.
- plėtinys (angl. *Ext plugin*) – suteikia programuotojui galimybę pakeisti bet kurią „Liferay“ klasę naudotojo aprašyta. Tai yra galingiausia, tačiau ir labiausiai linkusi sukelti klaidas, priemonė, todėl ją reikia naudoti tik tuomet, kai nepakanka kitų priemonių galimybių.
- portalo įskiepių išdėstymo šablonas (angl. *Layout template*) – leidžia koreguoti portalo įskiepių išdėstymą.

„Liferay“ yra platinamas su keletu populiariausių aplikacijų serverių („Jboss“, „Apache Tomcat“ ir kt.). PEPS diegiamas kartu su „Apache Tomcat“ dėl paprastumo ir patikimumo [3][11].

2.5.1.2. „Apache Tomcat”

„Apache Tomcat” („Jacarta Tomcat” arba tiesiog „Tomcat”) yra atviro kodo Java kalba parašytų aplikacijų konteineris, suteikiantis Java HTTP serverio aplinką. Sukurta „Apache Software Foundation“ (ASF). Šiuo metu jis yra vienas labiausiai naudojamų atviro kodo paketų pasaulyje bei aktyvia bendruomene prisidedančia prie tolesnio „Apache Tomcat” vystymo [1].

„Apache Tomcat” yra solidus nepriklausomas WEB serveris ir aplikacijų aplinka. Tačiau jo WEB serveris neturi tokio išplėtoto funkcionalumo kaip, pavyzdžiui, labiau įsitvirtinę WEB serveriai, tokie kaip „Apache web server“ (pavyzdžiui „Tomcat“ neturi didelio sąrašo papildomų modulių). Tačiau „Tomcat“ yra nemokama ir atviro kodo programinė įranga, ir daugybė talentingų žmonių prisideda prie jo augimo.

Kadangi „Tomcat“ yra parašyta Java programavimo kalba, dalis žmonių mano, jog reikalingos išsamios Java kalbos žinios norint juo naudotis, tačiau tai netiesa. Nors reikia žinoti Java norint pakeisti „Tomcat“ vidinį funkcionalumą, tačiau to nereikia norint juo naudotis.

2.5.2. Programiniai karkasai

Programinių karkasų (angl. *Framework*) pasirinkimas yra labai platus, net ir susiaurinus paiešką pagal programavimo kalbą. Visi skiriasi bendruomenės dydžiu, siūlomu lankstumo lygiu ir pan. Žemiau esančioje lentelėje pateikiamas populiariausių Java programinę kalbą naudojančių karkasų palyginimas (12 lentelė). Lyginama pagal šiuos kriterijus:

- AJAX naudojimas.
- Lokalizacijos palaikymas – internetinės aplikacijos dažnai kuriamos su galimybe pasirinkti kalbą, kuria atvaizduojamas tekstas, todėl toks palaikymas yra naudingas.
- Testavimas – ar palaiko naudotojo sąsajos testavimą.
- Apsauga – jei karkasas turi integraciją su apsaugos moduliais, tuomet reikia mažiau kodo ir konfigūracijos pačioje aplikacijoje.
- Šablonai – ar karkasas palaiko šabloninius komponentus.
- Formų tikrinimas – kadangi internetinė aplikacija dažnai sudaroma iš vienokių ar kitokių formų, jų reikšmių tikrinimas yra viena pagrindinių dalių [8].

12 lentelė. Programinių karkasų palyginimas

Projektas	Ajax	Lokalizavimas	Testai	Apsauga	Šablonai	Formų tikrinimas
Apache Click	jQuery	Taip	Taip	įdiegiama	Velocity, JSP	Taip
Apache Sling	Taip	Ne	Ne	Taip	Taip	Ne
Apache Struts	Taip	Taip	Unit tests	Ne	Taip	Taip
Apache Tapestry	Prototype, jQuery	Taip	Selenium TestNG JUnit	Spring Security, Shiro	Taip	JSR 330 (Bean Validation)
Apache Wicket	Priedai: YUI, Ext JS, kt.	Taip	Per plėtinius	Taip	Taip	Taip
JavaServer Faces	Taip	Taip	JUnit	Taip	Facelets, JSP	Taip
JBoss Seam	Taip	Taip	JUnit, TestNG	JAAS integracija, Drools, Hibernate Filters, OpenID, CAPTCHA	Facelets	Hibernate Validator
Jspx-bay	Taip	Ne	Ne	JAAS integration	Master-content pages	Taip
JVx WebUI	Taip	Taip	JUnit	Taip	Single sourcing	Įdiegiami
Spring	Taip	Taip	Taip	Spring Security	JSP, Commons Tiles, Velocity, Thymeleaf, kt.	Commons validator, Bean Validation
Stripes	Taip	Taip	Taip	karkaso praplėtimas	Taip	Taip
Vaadin	GWT	Taip	Taip	Ne	Taip	Taip
WebObjects	Taip	Taip	WOUnit (JUnit), TestNG, Selenium	Ne	Taip	Taip
ZK	jQuery	Taip	JUnit, ZTL	Spring Security	Macro komponentai	klientinėje dalyje, serveryje

Iš palyginimo duomenų išsiskiria „Apache Wicket“ ir ZK karkasai. Pirmoji PEPS portalo paslaugų administravimo modulio versija buvo kuriama su „Apache Wicket“, vėliau pereita prie „ZK“.

2.5.2.1. „Apache Wicket”

„Apache Wicket” (arba tiesiog „Wicket“) yra atviro kodo, lengvas, komponentų pagrindu sukurtas Java WEB karkasas kuriame panaudotas „Java Swing“ įvykių pagrindo (angl. *Event-based*) programavimo modelis internetinių aplikacijų kūrimui. Komponentiniai karkasai (angl. *Component based framework*) yra traktuojami kaip Java WEB programavimo ateitis, ir „Wicket“ yra vienas iš pirmaujančių produktų šioje aplinkoje. „Wicket“ siekia švariai atskirti HTML puslapio dizainerio ir Java programuotojo roles, palaikydama paprastus HTML šablonus, kurie gali būti keičiami, peržiūrimi ir vėliau patikslinti naudojantis standartinius WYSIWYG (angl. W*hat You See Is What You Get*, „ką matai ta ir gauni“) HTML dizaino įrankius [15].

„Wicket“ priešinasi HTTP būsenos nebuvimui pateikdama statiškus komponentus, taip padidindama produktyvumą. Jis turi konkrečią architektūrą ir skatina naudoti orientuotą į objektus (angl. *Object-oriented*) programavimą.

„Wicket“ labai skiriasi nuo veiksmas-užklausa (angl. *Action-request*) pagrindu parašytų bazių, tokių kaip „Struts“, „WebWork“ ar „Spring MVC“, kur formos siuntimas galiausiai paverčiamas vienu veiksmu. Čia naudotojo veiksmas aplikacijoje dažniausiai sukelia įvykį viename iš formos komponentų, kuris savo ruožtu atsako įvykių „klausantiems“ (angl. *Event listeners*). „Wicket“ yra panašus į „Tapestry“, „JSF“ ir „ASP.NET“. Galiausiai tokios bazės kaip „Struts“ davė pradą koncepcijai WEB MVC (angl. *Model-View-Controller*), kur „Wicket“ atneša labiau žinomą programavimo patirtį į internetą [5].

Šis karkasas, nors ir pažangus, yra labai glaudžiai susijęs su HTML. Tai jį daro mažiau lankstų, ir kuriant paslaugų formų administravimo komponentą teko parašyti daug kodo, kuris įgyvendintų gana paprastą funkcionalumą. Vėliau buvo nuspręsta pereiti prie „ZK“, labiau į JavaScript orientuoto internetinio karkaso.

2.5.2.2. „ZK“

Ne taip kaip daugelis Ajax pagrindu dirbančių karkasų, „ZK“ nereikalauja jokių žinių apie JavaScript. Tai pasiekiam dėl „ZK“ vidinio variklio, kuris generuoja visą reikalingą JavaScript kodą. Tam, kad sėkmingai pradėti naudoti „ZK“ tereikia šiek tiek žinoti apie HTML.

Aplikacijų kūrimo procesą sutrumpina naudojama puslapio struktūros užrašymo kalba ZUML (angl. *ZK User Interface Markup Language*). Jos pagalba įvarius komponentus galimą aprašyti viena eilute, lygiai taip pat kaip ir rašant HTML.

„ZK“ karkasas eina į serverį orientuotu keliu (angl. *Server-centric approach*), t.y. turinio iš komponentų sinchronizacija ir įvykių valdymas tarp klientų (naudotojų) ir serverio yra daromas vidinės logikos, ir AJAX kodas yra „nematomas“ aplikacijos kūrėjams. Taip užtikrinama, kad

galutiniai naudotojai gaus pilną ir funkcionalią naudotojo sąsają, kuri primena stacionarių kompiuterių programas, o programuotojams tai neprideda didesnio sudėtingumo.

Dėl savo orientavimo į JavaScript ir lankstumo programuotojo atžvilgiu, „ZK“ buvo tinkamiausias sprendimas PEPS portalui kurti. Technologiškai sudėtinguose moduliuose, tokiuose kaip paslaugų formų generatorius, „ZK“ leido lanksčiai dirbti su komponentais, nerašant klientinės dalies kodo. To nebūtų galima pasiekti kitais populiariais karkasais, tokiais kaip „Apache Wicket“ [2][16].

2.5.3. Google žemėlapių paslaugos

PEPS portale viena iš pagrindinių paslaugų yra pranešimas policijai. Norint šią paslaugą užsakyti, vienas iš privalomų duomenų yra tiksli vieta. Tam, kad naudotojui užsakančiam paslaugą būtų kuo paprasčiau ją nurodyti, reikalingas komponentas, kuris leistų tai padaryti greitai ir patogiai.

Adresas, iš programuotojo pusės, yra sudėtingas, kelių dalių objektas. Dažniausiai jį sudaro atskiri komponentai, tokie kaip valstybė, miestas, gatvė ir t.t. Adresas traktuojamas kaip užpildytas, jei yra visi laukai, bei iš jų galima nustatyti konkrečias koordinates.

Kadangi pildyti keletą laukų nėra patogu, pasitelkiamos žemėlapių paslaugos. Vienas iš lyderių šioje srityje yra Google, teikdamas „Google Maps API“. Kuriant epolicija.lt portalą buvo naudojama trečioji (V3) versija [6].

„ZK“ karkasas tiesiogiai integruotas su „Google Maps“, todėl turi paruoštą žemėlapio komponentą naudojimui. Žemėlapio komponente, vartotojas gali naršyti visą pasaulio žemėlapį ir pasirinkti konkrečią vietą. Iš pasirinktų koordinačių gaunamas pilnas vietovės adresas.

Taip pat galimas ir atvirkštinis procesas, pagal pateiktus adreso duomenis – valstybė, šalis, gatvė ir t.t., gaunamos potencialios (apytikslės) koordinatės (angl. *Reverse geocoding*). Šis servisas suteikia galimybę integruoti žemėlapio ir adreso komponentus, t.y. pasirinkus adresą – parodyti vietą žemėlapyje bei pasirinkus vietą žemėlapyje užpildyti adreso laukus [12].

3. EPOLICIJA.LT NAUDOTOJO SĄSAJOS REALIZACIJA

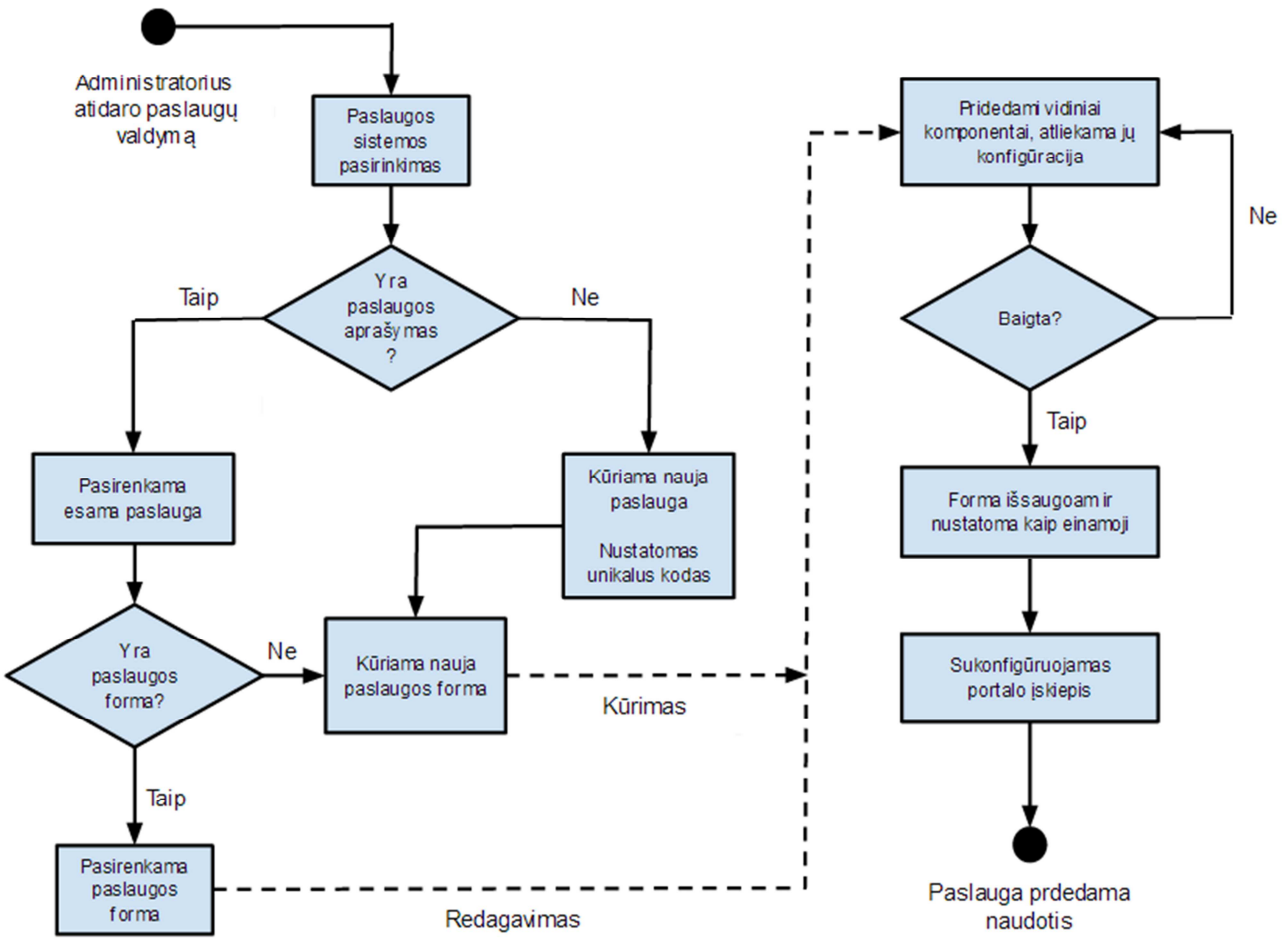
3.1. Sistemos administratoriaus dalis

Administratoriaus dalis epolicija.lt portale sudaro didelę dalį visos aplikacijos, kadangi jos pagalba valdomas visos teikiamos paslaugos, sukuriamos paslaugų įvedimo formos, valdomas apklausų kūrimas, analizuojami ir peržiūrimi statistiniai duomenys.

Asmuo, administruojantis PEPS portalą, esant poreikiui sukurti ar pakeisti paslaugą, turi atlikti šiuos veiksmus (eilės tvarka):

1. Prisijungti prie epolicija.lt portalo (žr. Prisijungimas prie administravimo sąsajos)
2. Pereiti į paslaugų administravimo puslapį (žr. Paslaugų administravimo puslapis).
3. Sukurti arba pasirinkti paslaugų sistemą bei sukurti paslaugą, t.y. įvesti pagrindinius duomenis, paslaugos aprašymą bei suteikti jai unikalų identifikavimo kodą (žr. Naujos paslaugų sistemos ir paslaugos sukūrimas, keitimas ir šalinimas).
4. Pridėti (nebūtinai) paslaugos įvedimo laukų šablonus (žr. Paslaugos duomenų įvedimo formos laukų šablonai).
5. Sukurti paslaugos užsakymo duomenų priėmimo formą (žr. Paslaugos užsakymo duomenų formos sukūrimas, keitimas ir šalinimas).
6. Sudėti reikiamus komponentus į paslaugos užsakymo formą, pasirinkti ar forma bus apsaugota nuo automatizuoto užsakymo ar ne, pritaikyti reikiamą stilių (žr. Formos nuostačių langas).
7. Pridėti duomenų formą į naują portalo puslapį, nurodyti unikalų formos ID portalo įskiepio parametruose.

Veiksmų schema parodyta žemiau (8 pav.).



8 pav. Paslaugos formos kūrimas

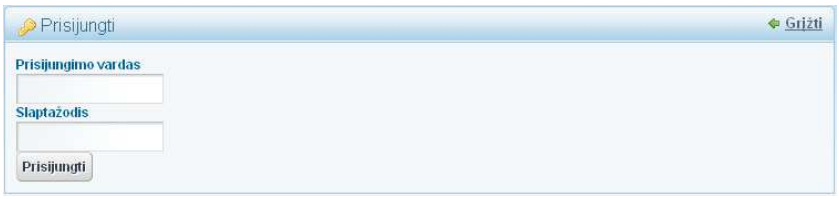
Po šių veiksmų, jei paslauga yra galiojanti (nustatoma pagal parametrus „galioja nuo“ ir „galioja iki“), vartotojai gali ja naudotis ir užsakyti paslaugas. Suvedus privalomus duomenis, jie bus siunčiami per bendrą PEPS mechanizmą, kur bus apdorojami ir persiunčiami į kitas sistemas.

Diegiant naują paslaugą PEPS portale, svarbu paruošti paslaugos užsakymo mechanizmą PUVS sistemoje, kitaip pranešimas bus nuskaitytas ir grąžintas klaidos pranešimas.

3.1.1. Prisijungimas prie administravimo sąsajos

Norint prisijungti prie epolicija.lt administravimo sąsajos reikia paspausti ant prisijungimo mygtuko ir į atsiradusią formą įvesti savo prisijungimo vardą ir slaptažodį (9 pav.).

Administravimo sąsają mato tik tas vartotojas, kuris turi administravimo teises. Šios teisės priskiriamos pagal suteiktą rolę. Portalo vartotojas gali turėti vieną iš dviejų galimų rolių: administratoriaus arba naudotojo rolę. Vartotojas turintis administratoriaus rolę, gali daryti struktūrinius pakeitimus, o naudotojo rolę – tik peržiūrėti informaciją bei naudotis viešai prieinamu funkcionalumu.



9 pav. Prisijungimo forma

3.1.2. Pradinis portalo administravimo sąsajos puslapis

Sėkmingai prisijungus prie portalo, matomas pradinis administravimo sąsajos puslapis (10 pav.). Kairėje pusėje yra pasirinkimų meniu, o dešinėje pusėje - pateikiama informacija, priklausomai nuo einamojo puslapio. Virš kairiojo meniu ir portalo turinio yra meniu juosta, kurioje yra nuorodos į kitus puslapius.

Viršutinė juosta suteikia pagrindinius „Liferay“ portalo funkcionalumus, kurie glaudžiai integruojasi su vaizduojamu turiniu. Naudojantis šia įrankių juosta, galima pridėti naujus portalo komponentus arba administruoti esamus (11 pav.). Paspaudus ant punkto „Add“, gauname sąrašą galimų komponentų, kuriuos galima pridėti į einamąjį puslapį. Šiame sąraše turime ne tik pagal nutylėjimą iš „Liferay“ konteinerio gaunamus komponentus, bet ir tuos, kurie buvo sukurti atskirai bei pridėti konfigūracijos pagalba.

Pagrindiniai „Liferay“ suteikiami komponentai:

- Internetinio turinio portalo įskiepis.
- Paprastų duomenų surinkimo formų portalo įskiepis.
- Puslapių reitingavimo portalo įskiepis.
- Apklausų rengimo komponentas.

Pagrindiniai PEPS portalo įskiepiai:

- Paslaugų administravimo portalo įskiepis
- Paslaugos formos atvaizdavimo portalo įskiepis.
- Naudotojo užsakytų paslaugų peržiūros portalo įskiepis.

10 pav. Pradinis epolicija.lt portalo administravimo sąsajos puslapis

11 pav. Įrankių juosta

3.1.3. Paslaugų administravimo puslapis

Norint valdyti teikiamas paslaugas, reikia pereiti prie paslaugų administravimo puslapio. Paslaugų administravimo puslapis leidžia patogiai ir greitai kurti naujas paslaugas. Šioms paslaugoms kuriamos duomenų įvedimo formos, kurias galima modifikuoti, kopijuoti ar ištrinti. Nedarant didelių pakeitimų, paslaugas galima išjungti ar laikinai sustabdyti jų naudojimą.

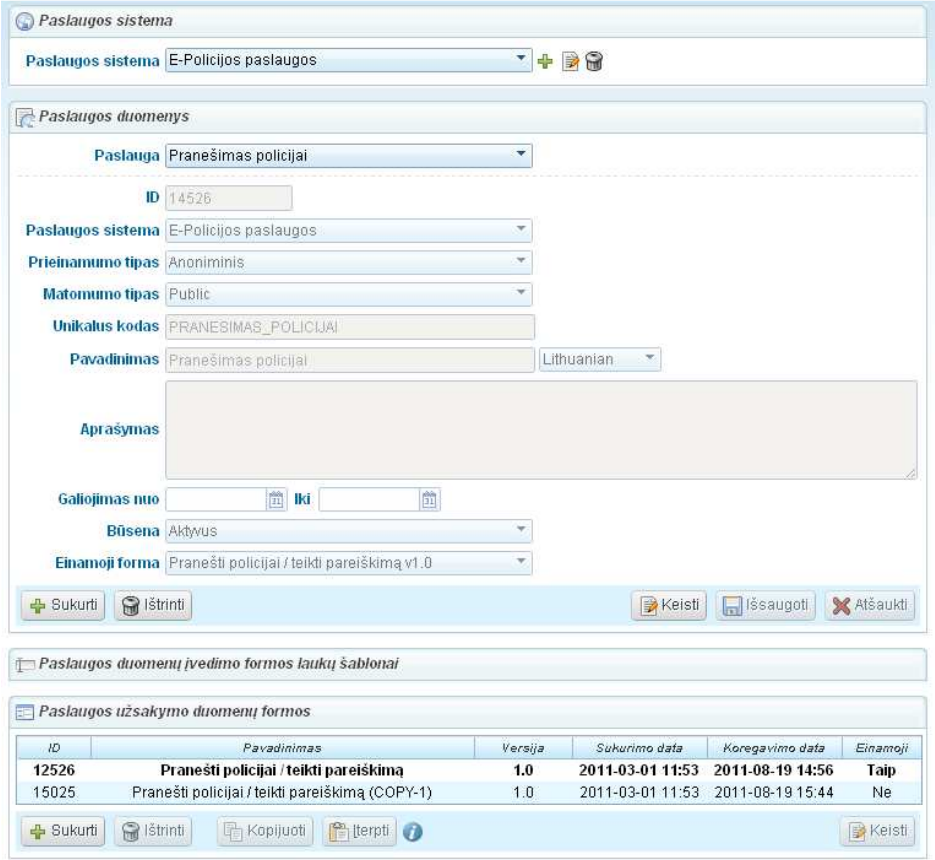
Paslaugų administravimo komponentas yra centrinė viso portalo dalis, apie kurią siejasi visos paslaugos. Šis komponentas realizuotas kaip ZK programiniu karkasu grindžiamas portalo įskiepis, t.y. programinis modulis, kurį galima talpinti į bet kurį puslapį keletą kartų. Jį, pasinaudojus

„Liferay“ funkcionalumu, galima prisidėti į bet kurią puslapio vietą, apriboti ilgį, plotį. Kadangi tai nėra viešai prieinama portalo dalis, šis komponentas patalpintas puslapyje, kuris nematomas eiliniam vartotojui.

Prie PEPS paslaugų administravimo puslapio pereinama paspaudus mygtuką „Paslaugų administravimas“, esantį pagrindiniame meniu kairėje pusėje (meniu nuoroda į šį puslapį rodoma tik prisijungusiam vartotojui su administratoriaus teisėmis) (12 pav.).

Komponentas išskirstytas į keturias pagrindines dalis:

- Paslaugos sistemos pasirinkimas/administravimas.
- Paslaugos duomenų informacija.
- Paslaugų duomenų įvedimo formos laukų šablonų valdymas.
- Paslaugos užsakymo formos.



12 pav. Paslaugų administravimo langas

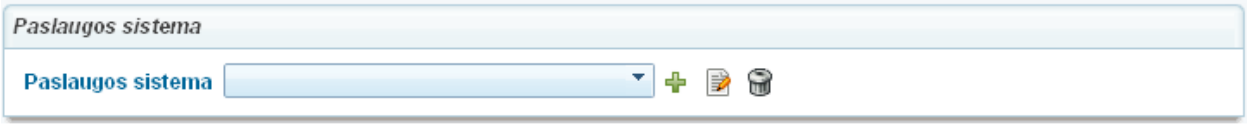
Pagal šias skiltis, vartotojas (administratorius) gali atlikti šiuos veiksmus:

1. Sukurti paslaugos sistemą arba, kuriant naują paslaugą, pasirinkti esamą.
2. Sukurti paslaugą, ją aprašyti, nustatyti jos parametrus.
3. Įvesti paslaugai duomenų įvedimo formų laukų šablonus.
4. Sukurti duomenų įvedimo formą ir ją priskirti paslaugai kaip einamąją formą.

3.1.4. Naujos paslaugų sistemos ir paslaugos sukūrimas, keitimas ir šalinimas

Paslaugos sistema – tai panašaus konteksto paslaugų apjungimas į vieną grupę. Paslaugos grupuojamos tam. kad suskirstyti skirtingas paslaugas pagal konkrečius kriterijus, pavyzdžiui paslaugos pobūdį, sąsają su specifine sistema ir t.t. Suskirsčius paslaugas į grupes, jas lengviau rasti ir administruoti (13 pav.).

Paslaugų sistemų valdymui skirti trys mygtukai (13 lentelė), kuriais galima pridėti, keisti ir pašalinti paslaugų sistemas. Jei bandoma pašalinti paslaugų sistemą, su kuria jau yra susietų paslaugų, sistema praneša jog to negalima padaryti kol nėra pašalintos individualios paslaugos.



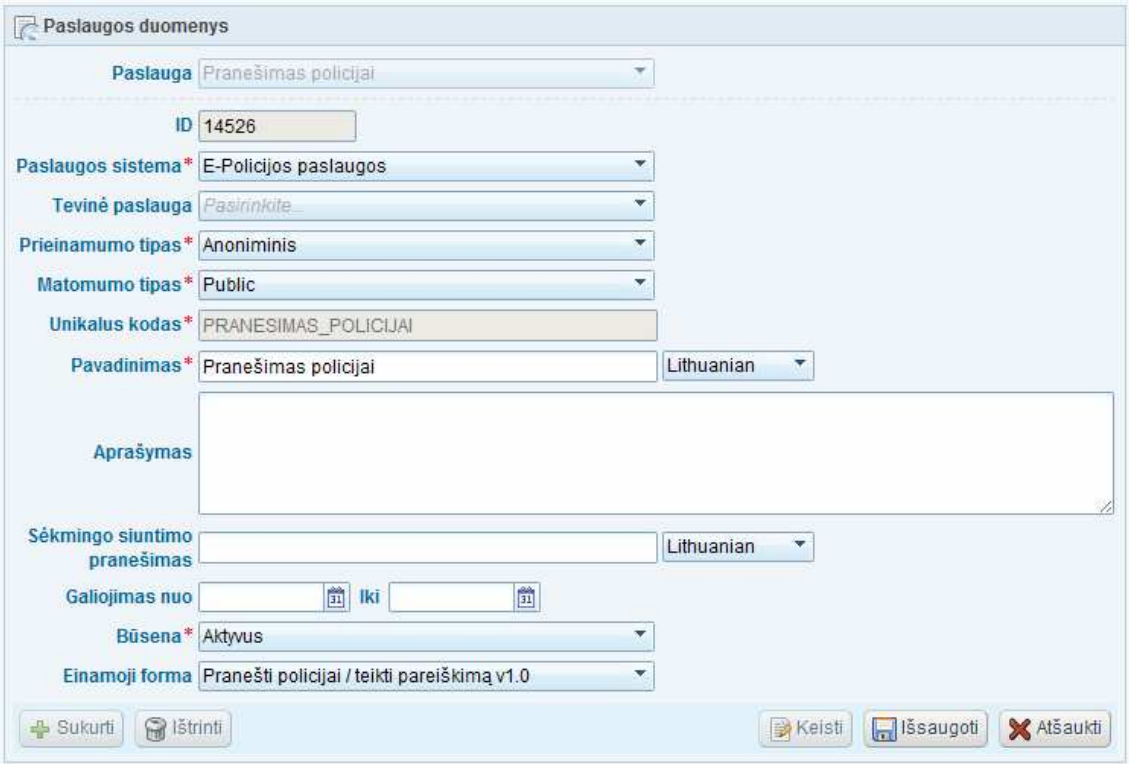
13 pav. Paslaugos sistema

13 lentelė. Paslaugos sistemos administravimo mygtukai

Mygtukas	Aprašymas
	Sukurti naują paslaugos sistemą
	Redaguoti pasirinktą paslaugos sistemą
	Pašalinti pasirinktą paslaugos sistemą

Norint sukurti naują paslaugą, pirmiausia reikia pasirinkti sistemą, su kuria ši paslauga bus susieta. Sukurti naują paslaugą, nepriskyrus jai paslaugos sistemos, negalima. Sukūrus paslaugos sistemą ir ją pasirinkus, aktyvuojasi mygtukas „Sukurti“, esantis paslaugų duomenų dalies apačioje (14 pav.). Jį paspaudus, aktyvuojasi visi laukai, kuriais aprašoma nauja paslauga. Paslaugos atributų sąrašas pateikiamas lentelėje žemiau (14 lentelė).

Paslaugos keitimo ar šalinimo atveju, taip pat reikia pasirinkti paslaugos sistemą. Tuomet matomos jai priklausančios paslaugos. Pasirinkus paslaugos sistemą ir reikalingą paslaugą, aktyvuojasi mygtukai „Keisti“ arba „Ištrinti“, kurie atitinkamai leidžia pakeisti arba ištrinti paslaugą. Paslaugos ištrinti (analogiškai kaip ir su paslaugų sistema) negalima, jei paslauga turi bent vieną, jai sukurtą, duomenų pateikimo formą. Norint ištrinti paslaugą, pirmiausia reikia ištrinti visas duomenų įvedimo formas. Toks susiejimas apsaugo nuo netyčinio, aktyvios paslaugos panaikinimo.



14 pav. Paslaugos duomenys

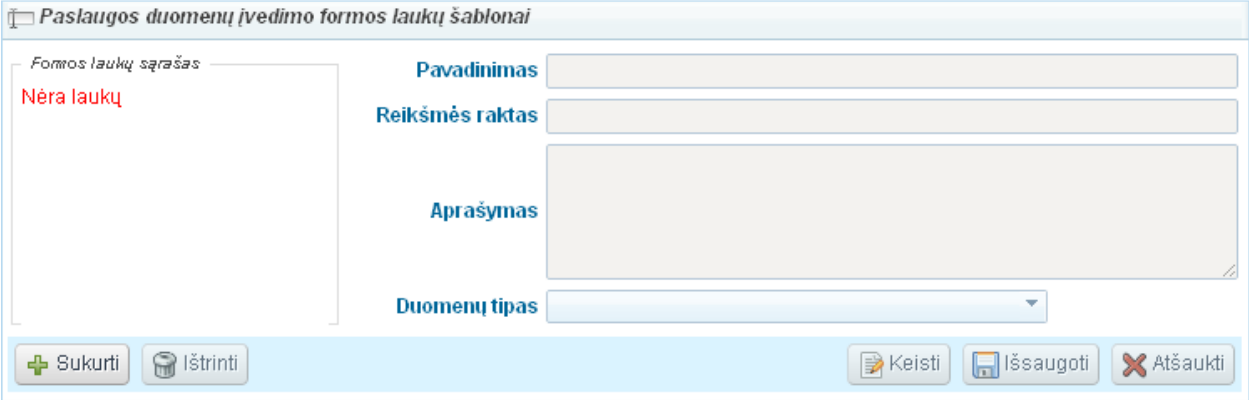
14 lentelė. Paslaugos duomenų laukų aprašymas

Lauko pavadinimas	Aprašymas
Paslauga	Einamoji (pasirinkta) paslauga.
ID	Paslaugos ID numeris (sutampa su duomenų lentelės, atitinkančios paslaugą, pirminiu lauku). Sukuriamas automatiškai.
Paslaugos sistema	Paslaugos sistemų sąrašas, pasirenkama atitinkama sistema, kuriai bus sukuriama paslauga.
Tėvinė paslauga	Paslauga, su kuria susijusi esama paslauga. Visos vaikinės paslaugos yra įtraukiamos į tėvinės paslaugos statistiką ir užklausas.
Prieinamumo tipas	Nurodo ar paslauga bus prieinama anonimiškai, ar bus naudojami asmeniniai duomenys.
Matomumo tipas	Nurodoma ar visa su paslauga susijusi informacija bus rodoma naudotojui.
Unikalus kodas	Unikalus paslaugos kodas.
Pavadinimas	Paslaugos pavadinimas (vedamas skirtingomis kalbomis, kurios yra palaikomos „Liferay“ portalo). Rodomas vartotojui.
Sėkmingo siuntimo pranešimas	Tekstas, rodomas vartotojui, sėkmingai išsiuntusiam užklausą konkrečiai paslaugai.
Aprašymas	Paslaugos aprašymas.
Galiojimas	Nurodomas paslaugos galiojimo laikotarpis. Jam pasibaigus (arba dar neprasidėjus), šia paslauga pasinaudoti negalima. Skirta apriboti tam tikros paslaugos tiekimą.
Būsena	Nurodo ar šiuo metu galima naudotis paslauga.
Einamoji forma	Forma, kuri bus susieta su šia paslauga ir bus rodoma naudotojui.

3.1.5. Paslaugos duomenų įvedimo formos laukų šablonai

Dažnai, kai vienai paslaugai yra sukurta daug formų, yra patogu naudotis formos laukų šablonais (15 pav.). Formos laukų šablonai aprašo lauko tipą ir priskiria jam vidinį kodą. Vidinis kodas – tai sistemos, į kurią bus siunčiami duomenys, lauko kodas. Kuriant formą galima nurodyti, kad formos komponentas naudotų aprašytą lauko šabloną. Tokiu būdu užtikrinama, kad naudojamas laukas, turintis tą pačią prasmę, turės tokį pat vidinį raktą.

Vienas iš taikymo atvejų galėtų būti toks: registruojant eismo įvykį, visada turi būti nurodomi tokie pagrindiniai duomenys: automobilio numeris, savininko vardas, pavardė. Jeigu vienai paslaugai reikės sukurti daug formų, kurios naudos šiuos duomenis, tada šablonai užtikrins, jog ta pati reikšmė bus susieta su tuo pačiu kodu (15 lentelė).



15 pav. Paslaugos duomenų įvedimo formos laukų šablonai

15 lentelė. Šablonų laukų aprašymai

Laukas		Aprašymas
Pavadinimas		Šablono pavadinimas
Reikšmės raktas		Išorinėje sistemoje atitinkančio lauko kodas. Formoje naudojamas kaip vidinis kodas.
Aprašymas		Šablono aprašymas.
Duomenų tipas	**Taip / ne reikšmė**	Šablonas bus naudojamas *checkbox* komponentui
	Slankiojo kablelio skaičius	Šablonas bus naudojamas *doublebox* komponentui
	Sveikasis skaičius	Šablonas bus naudojamas *intbox* komponentui
	Objektas	Šablonas bus naudojamas *combobox* komponentams
	Tekstinė reikšmė	Šablonas bus naudojamas *textbox*, *radiogroup* komponentams
	Data	Šablonas bus naudojamas *datebox*, *dateTimeBox*, *timeBox* komponentams
	Reikšmių sąrašas	Šablonas bus naudojamas *listbox*, *checkgroup* komponentas
	Reikšmių rinkinys	Šablonas bus naudojamas *listbox* komponentui

3.1.6. Paslaugos užsakymo duomenų formos sukūrimas, keitimas ir šalinimas

Paslaugos kūrimo pabaigoje, reikalinga sukurti bent vieną formą duomenims priimti, t.y. sukurti paslaugos formą, kurią vartotojas galės užpildyti. Paslaugos forma yra kuriama vienai konkrečiai paslaugai, todėl atliekant bet kokius veiksmus su formomis, pirmiausia būtina pasirinkti paslaugos sistemą bei tikslią paslaugą. Galimi veiksmai su formomis: kurti, keisti, ištrinti. Taip pat sukurta galimybė formas kopijuoti ir įterpti šalia reikiamos paslaugos. Patogu tais atvejais, kai reikia kurti naują formos versiją, pasinaudojant jau esamos formos pagrindu (16 pav.).

Norint panaikinti formą reikia įsitikinti, jog nėra paslaugų užsakymų naudojantis ta formos versija. Jei tokių yra, formos panaikinti negalima (užtikrinamas su senomis formomis pateiktų prašymų palaikymas).

16 pav. Paslaugos užsakymo duomenų formos

3.1.7. Formos nuostačių langas

Paspaudus mygtuką „Sukurti“, pereinama prie formos kūrimo lango. Jame medžio struktūra formuojami elementai, pradedant nuo aukščiausių arba tėvinių (angl. *Parent*) ir baigiant žemiausiais arba vaikiniais (angl. *Child*) (17 pav.).

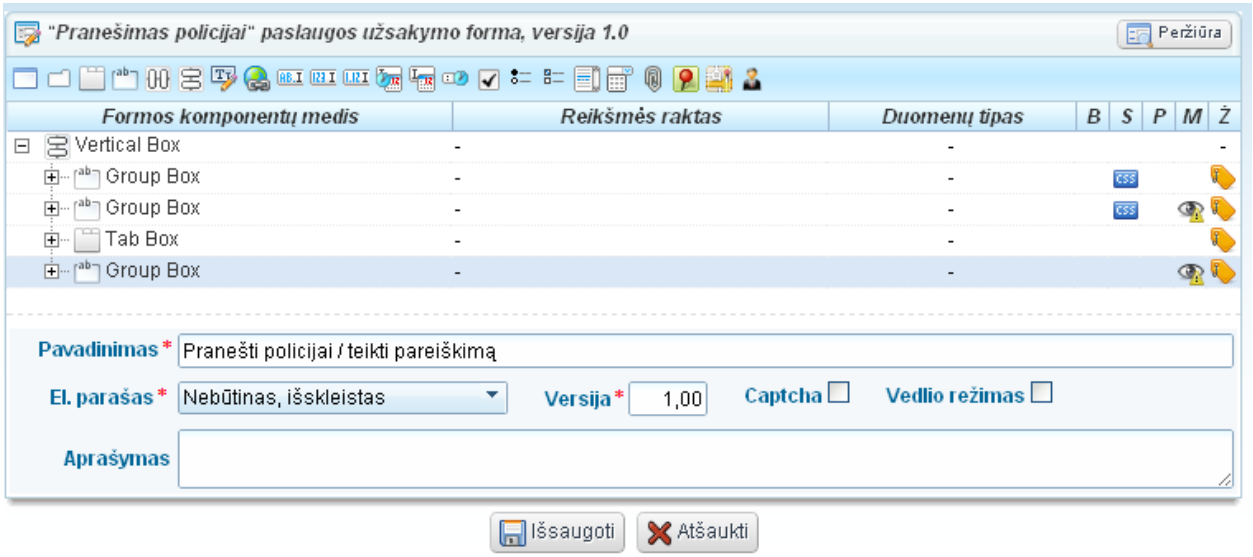
Pradinis komponentas, pradedantis formos struktūrą yra *VerticalBox* (arba *VBOX*), iš ZK karkaso atėjęs komponentas, kuris savo vaikinius komponentus sudeda vertikaliai. Kitus komponentus iš komponentų paletės parenka sistemos administratorius (16 lentelė).

Forma, be savo komponentinės struktūros, turi dar ir papildomų parametrų:

1. Pavadinimas – formos pavadinimas, naudojamas identifikuoti konkrečią formą visų paslaugos formų lentelėje.
2. El. parašas – pasirinkimas, nurodantis ar vartotojas privalo pateikiamus duomenis pasirašyti el. parašu.
3. Versija – skaičius reiškiantis formos versiją. Padeda identifikuoti eilės tvarką, pagal kurią buvo atnaujintos paslaugos duomenų pateikimo formos.

4. “Captcha”[1] – ar rodyti raidžių ir skaičių kombinaciją, kurią vartotojas turi įvesti prieš pateikdamas duomenis. Apsaugo nuo automatizuoto duomenų pateikimo.
5. Vedlio režimas – specialus pasirinkimas, sukuriantis vedlio tipo duomenų įvedimą (angl. *Wizard mode*). Šis pasirinkimas efektyvus tik tada, kai formos aukščiausieji komponentai yra *tabpanel* tipo. Tuomet kiekvienas iš šių komponentų tampa atskiru puslapiu, tarp kurių vaikštoma mygtukų pagalba. Skirtas sukurti kelių lygių duomenų priėmimą.

Sudėliojus komponentus ir paspaudus mygtuką „Peržiūra“, galima pamatyti kaip forma bus atvaizduota vartotojui. Atidaromas naujas langas, kuriame sugeneruojama pilna formos išvaizda, sukurta pagal pasirinktus komponentus. Peržiūros režime pelyte užvedant ant komponento, jis yra pažymimas tiek formoje, tiek pačiame komponentų medyje. Tokiu būdu galima atlikti pakeitimus ir matyti kaip jie įtakoja visą formos išvaizdą.



17 pav. Paslaugos duomenų įvedimo forma

Nauji elementai pridedami „tempimo ir metimo“ (angl. *Drag&Drop*) būdu, t.y. pasirinktas elementas tempiamas iki atitinkamos vietos formos komponentų medyje. Tempimo metu atsiranda mėlyna juosta pažyminti kur galima įterpti naują komponentą. Vieni komponentai gali talpinti savyje kitus komponentus (16 lentelė. Nr.: 1 – 6), kiti negali (16 lentelė. Nr.: 7 – 23) [16].

[1] captcha - angl. *Completely automated public Turing test to tell computers and humans apart*, raidžių ir skaičių kombinacija paveikslėlyje, kuriuos naudotojui reikia suvesti, norint toliau vykdyti su šiuo komponentu apsaugotą veiksmą [4].

16 lentelė. Paslaugos duomenų įvedimo formos komponentai

Nr.	Komponentas	Pavadinimas	Aprašymas
1.		**Panel**	Rėmeliu apjungia įvairius komponentus
2.		**Tab Panel**	Kitų komponentų talpykla, kortelės pavidalu
3.		**Tab Box**	Kortelių talpykla. Įgalina vedlio režimą
4.		**Groupbox**	Apjungia panašaus konteksto elementus į vieną grupę; vienu pelės spustelėjimu galima juos visus paslėpti
5.		**Vertical Box**	Eilučių talpykla, grupuoja eilutes į stulpelį
6.		**Horizontal Box**	Stulpelių talpykla, talpina elementus stulpeliais
7.		**Rich Text**	Komponentas tekstinio konteksto pavaizdavimui
8.		**Internet Link**	Nuorodos komponentas
9.		**Text Input**	Tekstinio įvedimo laukas
10.		**Integer Input**	Sveikųjų skaičių įvedimo laukas
11.		**Double Input**	Slankiojo kablelio skaičiaus įvedimo laukas
12.		**Date/Time Input**	Datos ir laiko įvedimo laukas, atvaizduojamas kalendorius ir laikas
13.		**Date Input**	Datos įvedimo laukas, atvaizduoja kalendorių
14.		**Time Input**	Laiko įvedimo laukas
15.		**Check**	Laukas, kurį galima pasižymėti varnele
16.		**Radio Group**	Sąrašas reikšmių, kur galima pasirinkti tik vieną reikšmę
17.		**Check Group**	Sąrašas reikšmių, kurias galima pasižymėti
18.		**List**	Sąrašas reikšmių pasirinkimui
19.		**Dropdown Input**	Pasirinktinio įvedimo laukas, pateikia išskleidžiamą sąrašą
20.		**Files Upload**	Bylų prisegimo komponentas
21.		**Address Input**	Adresų komponentė
22.		**Google Map**	*Google* žemėlapis
23.		**Legal Entity**	Juridinių asmenų paieškos komponentas

3.1.8. Formos komponentų parametrai

Kiekvienas iš formoje esančių komponentų turi tam tikrą kiekį parametrų, kuriuos formos kūrėjas gali keisti. Tikslus skaičius skiriasi nuo komponento tipo. Struktūriniai elementai jų turi mažiau, reikšmės įvedimo komponentai – daugiau. Valdant šiuos parametrus, komponentui galima pakeisti lauko reikšmės rodymo formatą, plotį ar net papildomą CSS stilių. Visų galimų parametrų sąrašas pateikiamas lentelėje žemiau (17 lentelė).

17 lentelė. Formos komponentų parametrai

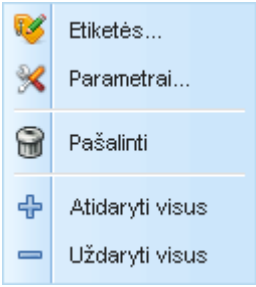
Lauko pavadinimas	Aprašymas
Komponentas	Komponento, kurio parametrai bus keičiami, tipas. Laukas neaktyvus, pateikiamas tik peržiūros režimu.
Formos lauko šablonas	Pritaikytas lauko šablonas.
Išskleistas	Nustato ar komponentas turi būti išskleistas pagal nutylėjimą. Taikomas komponentams, turintiems galimybę rodyti išskleistą ir suskleistą variantą.
Nerodyti peržiūroje	Nustato ar reikia rodyti komponentą (ir jo vaikus) formoje įvestų duomenų peržiūroje
Išorinis plotis	Nustato komponento *išorinio* konteinerio plotį pikseliais. Pagal nutylėjimą komponentai esantys *horizontaliame* konteineryje turi vienodą plotį. Šis parametras leidžia tai pakeisti.
CSS stilius	Komponento CSS stilius, leidžia tiksliai valdyti komponento atvaizdavimo parametrus
Turinio CSS stilius	Komponento turinio CSS stilius, leidžia valdyti komponento turinio atvaizdavimo parametrus
URL	Nuoroda į kitą puslapį, pvz. „*http://www.google.com*“
Išjungtas	Įjungus komponentas rodomas tik peržiūros režimu.
Išdėstymo tipas	Lauko etiketės (pavadinimo) išdėstymas.
Duomenų tipas	Komponento duomenų tipas.
Vidinis kodas	Vidinis komponento kodas, pagal kurį komponento laukų reikšmės bus susietos su išorinėmis sistemomis.
Reikšmė privaloma	Nurodoma ar laukas gali būti tuščias, ar vis dėlto reikšmė yra privaloma.
Pradinė reikšmė	Pradinė lauko reikšmė, kuri bus rodoma naudotojui.
Minimalus ilgis	Nurodo kokio minimalaus ilgio gali būti įvedama reikšmė.
Maksimalus ilgis	Nurodo kokio maksimalaus ilgio gali būti įvedama reikšmė.
Tikrinimo išraiška	Nustato komponente įvestos reikšmės tikrinimo naudojant „RegExp“[2] išraišką
Kelios eilutės	Nustato ar įvedamas tekstas gali turėti keletą eilučių
Eilučių skaičius	Nustato teksto įvedimo komponento matomų eilučių skaičių
Auto-plotis	Komponentas automatiškai parenka plotį, priklausomai nuo tėvinio komponento
Išorinis plotis	Nustato komponento *išorinio* konteinerio plotį pikseliais. Pagal nutylėjimą komponentai esantys *horizontaliame* konteineryje turi vienodą plotį. Šis parametras leidžia tai pakeisti.
Vidinis plotis	Komponento *vidinis* plotis pikseliais arba procentais. Daugelis komponentų susideda iš 3 dalių: *ŽYMĖS*, *ĮVESTIES* ir *PAGALBOS*. Šis parametras kontroliuoja *ĮVESTIES* dalies plotį.
Etiketės plotis	Nustato komponento etiketės ploti procentais arba pikseliais, pvz. 35%, 100px
Etiketės išdėstymas	Nustato komponento etiketės ploti procentais arba pikseliais, pvz. 35%, 100px

[2] RegExp – mechanizmas, skirtas išrinkti teksto dalis, atitinkančias tam tikrą formatą. Naudojama sulyginti ar vienas teksto fragmentas atitinka nurodytą formatą, pavyzdžiui ar pateiktas tekstas yra tinkamas el. pašto adresas [13].

17 lentelės tęsinys

Lauko pavadinimas	Aprašymas
Palėpti etiketę	Nustato ar komponento etiketė yra rodoma (savybė paveikia TIK etiketės tekstą, išdėstymas nepasikeičia)
Tik skaitymui	Nustato ar komponento etiketė yra rodoma (savybė paveikia TIK etiketės tekstą, išdėstymas nepasikeičia)
Išjungtas	Nustato ar komponentas yra "išjungtas"
Navigacijos indeksas	Nurodomas skaičius, pagal kurį pereinama prie kito komponento paspaudus „Enter“ klavišą.
Anksčiausia data	Nurodoma kokia gali būti datos lauko įvedama anksčiausia data. Pasirenkama iš reikšmių sąrašo.
Vėliausia data	Nurodoma kokia gali būti datos lauko įvedama vėliausia data. Pasirenkama iš reikšmių sąrašo.
Reikšmės formatas	Komponente vaizduojamos reikšmės formatas (pvz. datos/laiko arba skaičiaus)
Reikšmių sąrašas	Paprastas galimų reikšmių sąrašas, pvz. "Žymė-1=Reikšmė-1|Žymė-2=Reikšmė-2" ir t.t.
Orientacija	Komponento išdėstymas, savybės efektas priklauso nuo komponento
Keletas reikšmių	Nustato ar leidžiama pasirinkti daugiau kaip vieną reikšmę iš sąrašo
Klasifikatoriaus pavadinimas	Klasifikatoriaus lentelės pavadinimas, pvz. IVYKIO_TIPAS
Palaikomi failų tipai	Palaikomu failų tipų, atskirtų kablelių, sąrašas, pvz. "DOC, DOCX, XLS" ir t.t.
Integruoti su žemėlapiu	Nustato ar komponentas turi būti integruotas su *Google Maps* komponentu (jeigu jis yra)
Vidinių žymių plotis	Vidinių komponentų žymių plotis procentais
Matomi adreso laukai	Nustato adreso laukus kurie yra rodomi
Privalomi adreso laukai	Nustato adreso laukus kurių reikšmė yra privaloma
Nekeičiami adreso laukai	Nustato adreso laukus kurių reikšmė negali būti keičiama
Rodyti „valyti“ mygtuką	Nustato ar reikia rodyti visų adreso laukų duomenų išvalymo mygtuką
Integruoti su adresu	Nustato ar komponentas turi būti integruotas su adreso įvedimo komponentų (jeigu jis yra)
Žemėlapio plotis	Žemėlapio plotis pikseliais arba procentais (pvz. 75% arba 500px). Likusi vieta bus skirta adreso duomenų lentelei.
Žemėlapio aukštis	Žemėlapio aukštis pikseliais (pvz. 400px)
Rodyti duomenų lentelę	Nustato ar reikia rodyti einamosios vietos duomenų lentelę
Duomenų lentelės laukai	Nustato laukus, kurie turi būti matomi einamosios vietos duomenų lentelėje

Parametrų langas atidaromas spaudžiant dešiniu pelės klavišu ant medyje esančio komponento ir iš atsiradusio kontekstinio meniu pasirinkus „Parametrai...“ (18 pav.).

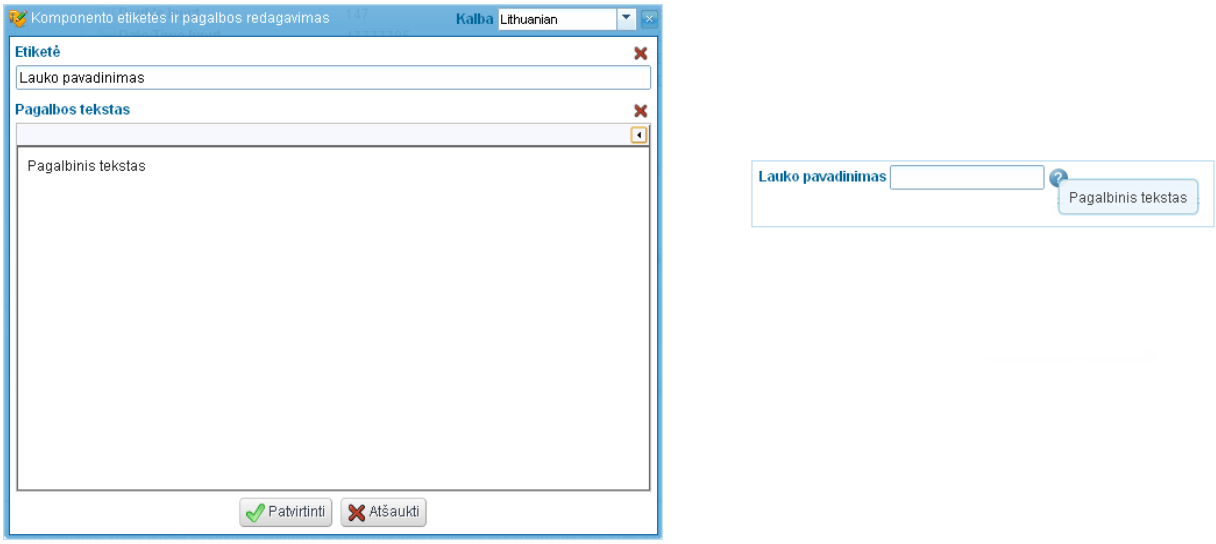


18 pav. Formos komponento kontekstinis meniu

Vieną komponentą sudaro trys pagrindiniai elementai, išdėstyti horizontaliai:

- Komponento pavadinimas
- Reikšmės įvedimo komponentas
- Pagalbos žymė

Dalis parametrų gali pakeisti šią struktūrą, pavyzdžiui, nerodyti komponento pavadinimo arba sumažinti plotį, kuriame rodomas pavadinimas. Pačio teksto konfigūravimas iškeltas į atskirą kontekstinio meniu dalį, pavadinimu „Etiketės...“. Atsidariusiame lange galima pakeisti šias tekstines reikšmes kiekvienai portalo palaikomai kalbai (19 pav.).



a) b)

19 pav. Komponento etikečių redagavimas: a) įvedimo langas b) rezultatas

Paslaugų formos komponentams vienas svarbiausių parametrų yra vidinis kodas. Kai naudotojas užpildo paslaugos formą, gauta informacija pagal vidinį kodą gali būti susieta su kitomis išorinėmis sistemomis.

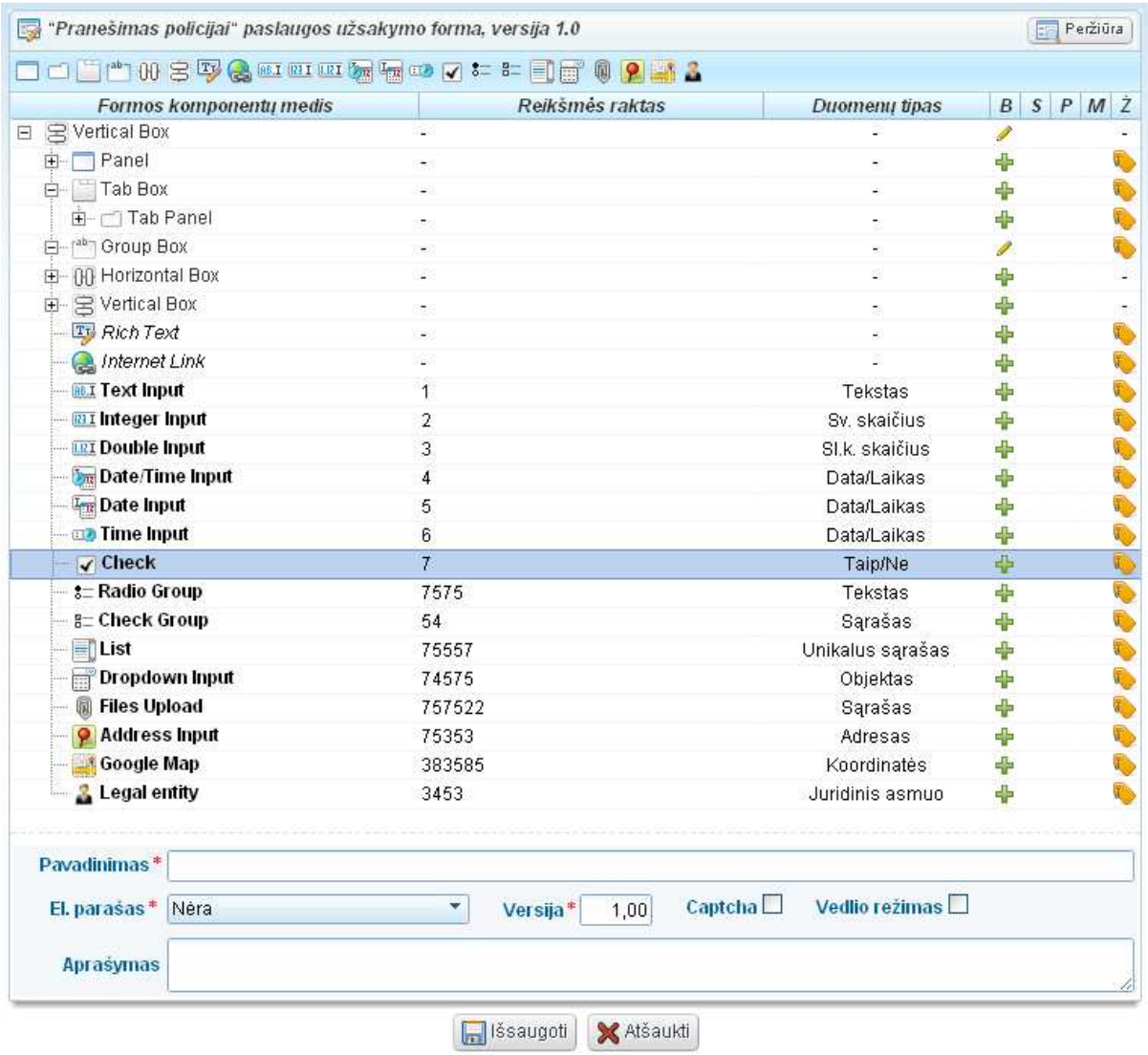
Portale sukurtoms paslaugoms vidinis kodas yra formuojamas sekančiu būdu:

„[sistemos pavadinimas].[paslaugos pavadinimas].[duomenų grupė].[duomenų pavadinimas]“

Išlaikant konkrečią pavadinimų konvenciją, užtikrinamas duomenų perdavimas bei teisingas sistemos veikimas.

3.1.9. Formos peržiūra

Sudėjus formos komponentus į medžio struktūrą, jame matome tik jos specifinę informaciją, t.y. komponento pavadinimą, reikšmės raktą, duomenų tipą, būseną ir žymę ar komponentui jau priskirta etiketė (20 pav.).

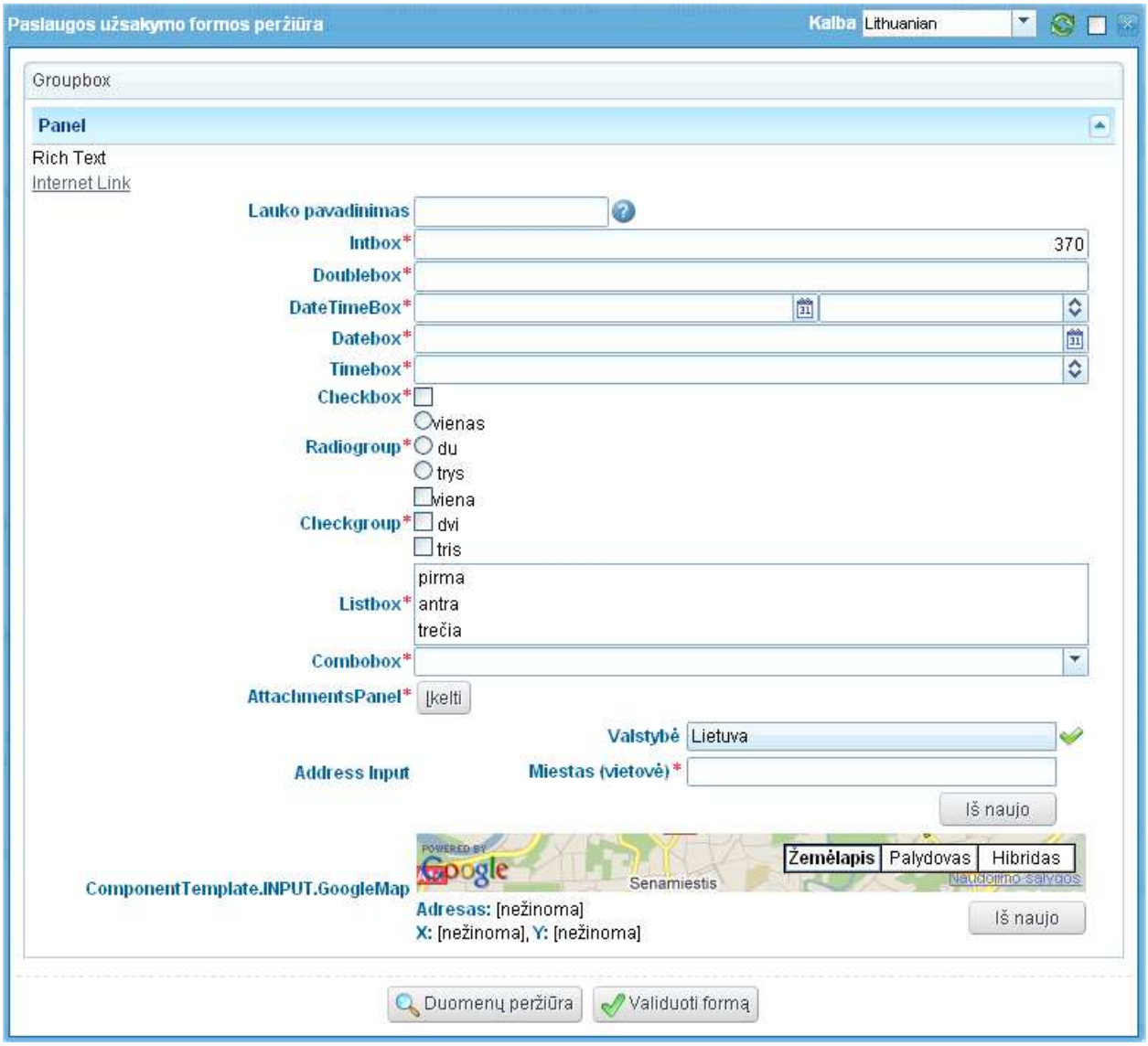


20 pav. Paslaugos duomenų formos struktūros medis

Paspaudus mygtuką „Peržiūrėti“, atidaroma sugeneruota duomenų įvedimo forma (21 pav.). Iš šio lango galima daryti du veiksmus – patikrinti suvedamus duomenis bei pamatyti kaip atrodo suvestų duomenų peržiūra.

Peržiūros metu matoma forma yra tokia, kokia bus matoma vartotojui. Pakeitus kalbą, galima matyti kaip forma atrodys su etiketėmis šiai kalbai. Tokiu būdu galima aptikti kuriems komponentams buvo nustatytas klaidingas tekstas ir kur jo visai nėra.

Atlikus pakeitimus komponentų medyje (pridėjus naujus komponentus ir pan.), pakeitus komponento parametrus, reikia paspausti mygtuką „atnaujinti“, kas pergeneruoja visą formą su atliktais pakeitimais.



21 pav. Paslaugos duomenų forma peržiūros režimu

Užpildžius duomenų formą testiniais duomenimis ir paspaudus „Validuoti formą“, sistema atlieka suvestų duomenų tikrinimą. Tai skirta patikrinimui ar sukurta forma teisingai tikrina duomenis ir neleidžia pateikti informacijos su neteisingomis ar trūkstamomis reikšmėmis. Paspaudus „Duomenų peržiūra“ mygtuką, atsidaro langas, kuriame pateikiami suvesti duomenys šioje formoje. Administratorius gali patikrinti ar formoje yra visi reikalingi laukai, o vartotojas (kuris mato šį vaizdą po duomenų pateikimo), gali patikrinti kokius duomenis jis nurodė.

3.2. Naudotojo dalis

Neprisijungęs vartotojas gali pasiekti tik viešai prieinamus puslapius bei naudotis anoniminėmis paslaugomis. Prisijungęs vartotojas mato visas paslaugas, bei gali sekti kiekvieno pranešimo būseną, bei peržiūrėti pateiktus duomenis.

3.2.1. Prisijungimas prie portalo

Prisijungimas nėra būtinas norint pasinaudoti PEPS portalo teikiamomis paslaugomis. Pateikti pranešimą arba pasinaudoti kita teikiama paslauga galima anoniminiu būdu. Identifikuoti save galima su prisegtu el. parašu pasirašytu dokumentu arba pasirašant patį siunčiamą pranešimą.

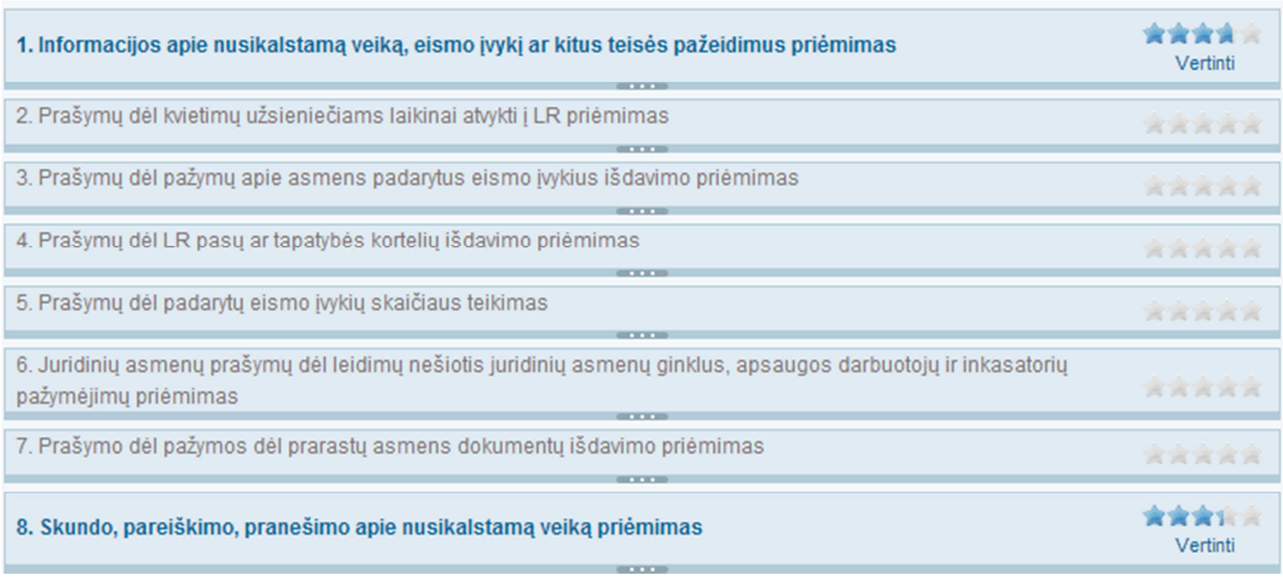
Prisijungę naudotojai turi galimybę sekti savo išsiųstų pranešimų būklę, tvarkyti naudotojo paskyros nustatymus ir greičiau užpildyti teikiamų paslaugų formas. Prisijungusiems naudotojams asmeniniai duomenys yra įvedami automatiškai, todėl užsakant paslaugas nereikia daug kartų suvedinėti tų pačių duomenų.

Prisijungti prie PEPS portalo galima paspaudus prisijungimo mygtuką, esantį pagrindiniame meniu (kairėje pusėje). Atsidariusiame puslapyje reikia pasirinkti tinkamą prisijungimo būdą. epolicija.lt teikia keletą prisijungimo galimybių, tarp kurių yra prisijungimas per banką.

3.2.2. Visos epolicija.lt paslaugos

Pasirinkus meniu punktą „Visos epolicija.lt paslaugos“ atidaromas puslapis su pateikta informacija apie visas policijos paslaugas, teikiamas elektroniniu būdu (22 pav.). Kiekvienai paslaugai sukurta lentelė (konfigūruojama administratoriaus), su pagrindine informacija. Šiame portalo įskiepyje (arba komponente), suteikta galimybė įvertinti teikiamą paslaugą nuo vienos iki penkių žvaigždučių. Neprisijungusiam vartotojui leidžiama įvertinti vieną kartą per mėnesį (tikrinama pagal IP adresą). Registruotas vartotojas gali įvertinti paslaugą kartą per mėnesį, tačiau jis gali keisti savo įvertinimą. Paslaugų įvertinimo statistikoje įskaitomas tik paskutinis (vėliausias) naudotojo įvertinimas. Vertinti galima tik aktyvias paslaugas.

Paslaugos, kurios šiuo metu yra viešai pasiekiamos, pateiktos ryškesniu šriftu. Pasiekiamumą įtakoja įvairūs veiksniai, pavyzdžiui pranešimų valdymo sistemos užimtumas, atliekami atnaujinimo, tvarkymo darbai arba tiesiog konkreti paslauga dar nėra realizuota.



22 pav. Teikiamų paslaugų sąrašas su vertinimo galimybe

3.2.3. Pranešimas policijai

Pati populiariausia paslauga, realizuota PEPS, yra pranešimas policijai. Viena iš šios paslaugos populiarumo priežasčių yra galimybė pranešti policijai neišeinant iš namų. Pranešimui išsiųsti tereikia užpildyti duomenis į administratoriaus sukurtą formą, nurodyti adresą, susijusį su įvykiu ir laukti rezultatų. Pranešimo forma pasiekiama paspaudus mygtuką „Pranešti policijai“.

Vartotojas yra nukreipiamas į puslapį, kuriame yra administratoriaus sukurta forma. Ją užpildžius, duomenys asinchroniškai siunčiami į pranešimų apdorojimo modulį, kuris vėliau atsiunčia pranešimus apie būsenų pasikeitimus. Pranešimo siuntimas ir būsenų valdymas realizuojamas per JMS eiles, į kurias dedami ir nuskaitomi duomenys.

Pranešimą galima nusiųsti anonimiškai, t.y. nenurodant asmeninių duomenų, tačiau visi privalomi laukai turi būti užpildyti.

Įvykio vietą galima nurodyti užpildžius įvykio adreso laukus, arba pasižymėjus įvykio vietą žemėlapyje. Adresas randamas pagal vietos koordinates, bei panaudojant „Google geocoding“ paslaugomis. Ryšys tarp įvedamo adreso ir nurodomos vietos žemėlapyje yra tiesioginis, t.y. įvedus tam tikrą adresą, jis bus atvaizduojamas žemėlapyje, o jeigu įvykio vieta nurodoma naudojantis žemėlapiu, tai atitinkamai užsipildys ir adreso laukai. Kuriant formą, adreso ir žemėlapio komponentai yra atskiri, tačiau turi parametrą, įgalinantį jų susietumą.

3.2.4. Naudotojo duomenys

Naudotojo paskyroje „Mano duomenys“, prisijungęs naudotojas gali pateikti kontaktinę bei papildomą informaciją, peržiūrėti savo duomenis. Esant reikalui, pavyzdžiui pasikeitus gyvenamajai vietai, juos gali pakeisti, papildyti. Naudotojo paskyra nėra sukurta naudojantis tuo pačiu

mechanizmu kaip ir duomenų rinkimo formos, vietoje panaudota „Liferay“ naudotojų valdymo sąsaja. „Liferay“ palaiko naudotojų valdymą bei jų prisijungimą, profilius, todėl tereikėjo naudotojo informaciją pritaikyti epolicija.lt vartotojui. Buvo pridėti tikslesni adreso laukai, informavimo duomenys.

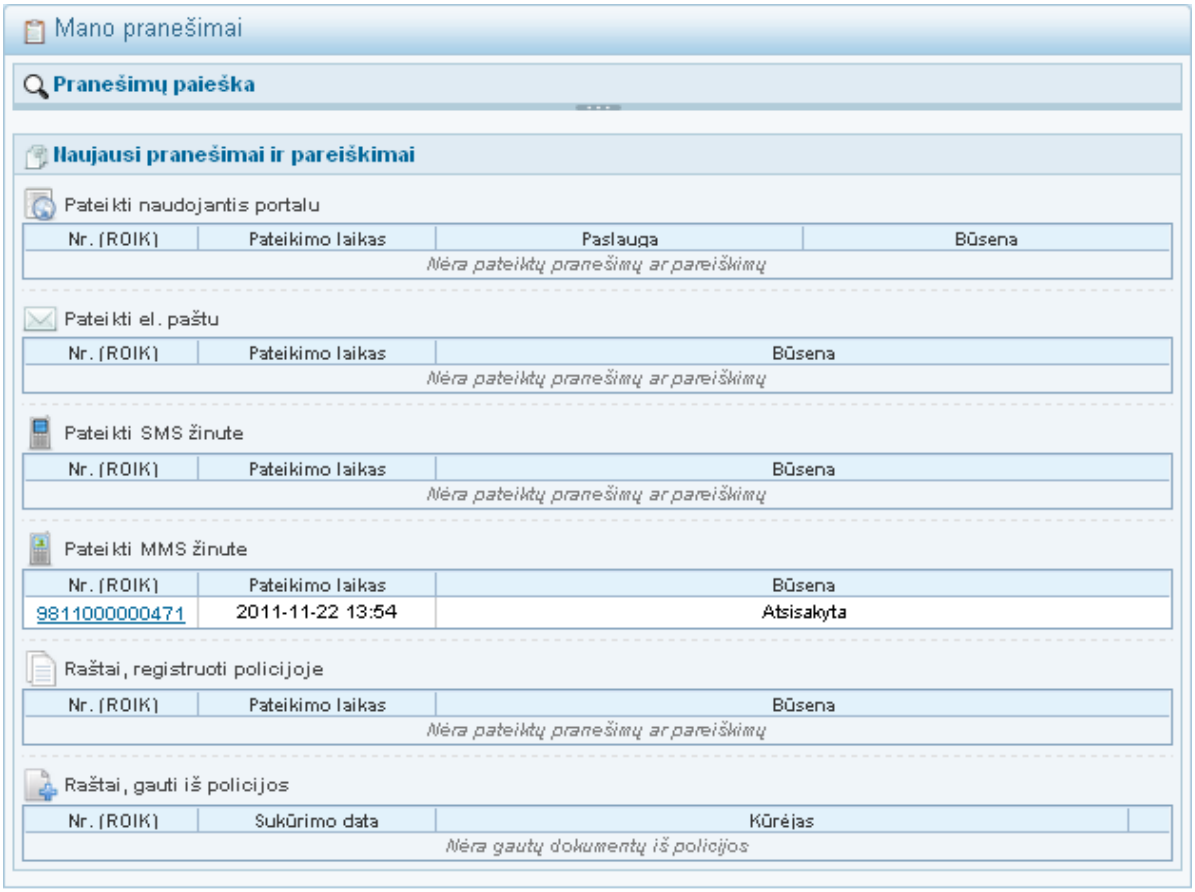
Naudotojas, norėdamas gauti pranešimus el. paštu arba SMS žinute, pirmiausia privalo įvesti savo el. pašto adresą ar mobilaus telefono numerį ir juos išsaugoti. Tada jam bus atsiųstas patvirtinimo kodas (į mobilųjį telefoną arba į el. paštą, priklausomai kuriuos duomenis įvedė). Jeigu per tam tikrą laiką patvirtinimo kodas nebuvo gautas, paspaudus ant užrašo „Naujas kodas“ bus sugeneruotas ir išsiųstas naujas patvirtinimo kodas. Kol mobilaus telefono numeris arba el. pašto adresas nebus patvirtinti, tol naudotojas negalės pasirinkti jokių nustatymų susijusių su šiais duomenimis.

3.2.5. Naudotojo pranešimai

Visos teikiamos paslaugos siejasi su naudotojo pranešimų portalo įskiepiu (23 pav.). Jis patalpintas puslapyje, pasiekiamame kai vartotojas yra prisijungęs. Pagal prisijungusį vartotoją jame rodomi visi, pagal pateikimo būdus išskirstyti pranešimai.

Galima atlikti paiešką tarp visų pateiktų pranešimų, o rezultatus naudotojas gali atidaryti peržiūros režimu, atsispausdinti. Taip pat pateikiama būsenų istorija.

Šis portalo įskiepis apjungia visus dinamiškai sukurtas paslaugų užsakymus į vieną centralizuotą vietą, kurioje naudotojas gali juos peržiūrėti bei atlikti kitus veiksmus. Norėdamas peržiūrėti detalesnę informaciją, naudotojas turi paspausti ant dominančio įrašo ROIK numerio.



23 pav. Prisijungusio naudotojo užsakytos paslaugos

4. REZULTATAI IR IŠVADOS

Portalo epolicija.lt kūrimas išsiskyrė nuo kitų taikomųjų programų kūrimo projektų savo unikalumu ir užsibrėžtais tikslais. Projekto metu buvo atlikta ne tik sistemos analizė, tačiau ir tinkamiausių technologijų tyrimas.

Buvo atliktas technologijų tyrimas pagal šias dvi grupes:

- internetinių taikomųjų programų pagrindas arba jų konteineriai;
- internetiniai karkasai, programiniai paketai.

Iš šių grupių išskirti sprendimai, kurie panaudoti epolicja.lt portale:

- **„Liferay“ taikomųjų programų konteineris**. „Liferay“ taikomųjų programų portalas parinktas dėl suteikiamos programinės bazės, t.y. integruoto turinio ir portalo įskiepių valdymo mechanizmo, naudotojų palaikymo. Nuo kitų portalų „Liferay“ išsiskyrė savo išbaigtumu ir saugumu. Nors buvo siūloma tiek mokama, tiek nemokama versija, buvo pasirinkta nemokama, nes jos visiškai pakanka visam reikiam funkcionalumui realizuoti.
- **„ZK“ internetinis karkasas.** „ZK“ išsiskyrė iš kitų programinių karkasų rinkoje savo panašumu į JavaScript karkasą, tačiau visiškai paslepiantį visą vidinę logiką ir netgi HTML elementus. Dėl savo lankstumo jis labiausiai tiko sudėtingai formų generavimo logikai. „Apache Wicket“ taip pat labai populiarus karkasas netiko PEPS portalo kūrimui dėl savo apriboto dinamiškumo, t.y. puslapio komponentų valdymo trūkumo. „Apache Wicket“ yra suderintas statiškoms internetinėms programoms kurti.

Dirbant su portalu ir kuriant portalo įskiepius, labai svarbu įvertinti suderinamumo problemas. Portalo įskiepiai, kurių veikimas labai glaudžiai susijęs su kliento pusėje veikiančia logika, labiausiai kertasi su pagrindine portalo sistemos veikla. Dėl šios priežasties, jei projektuojama sistema naudoja sudėtingą logiką, gali būti naudingiau kurti pagrindą nuo nulio, nei naudoti portalą. PEPS atveju, kai formos generuojamos dinamiškai, buvo susidurta su problemomis, kurias spręsti teko nagrinėti portalo išeities kodą. Nors buvo susidurta su problemomis, tačiau privalumai, naudojant portalą, ženkliai persveria neigiamus aspektus.

Vienas iš svarbiausių PEPS portalo komponentų, realizuotų praktinėje dalyje – **paslaugų duomenų priėmimo formų konstravimo modulis**. Naudodamasis šiuo komponentu, sistemos administratorius gali visiškai valdyti paslaugų duomenų priėmimo formas ir netgi kuriam laikui sustabdyti ar visai išjungti paslaugų tiekimą.

Toks dinamiškumas pasiektas su „ZK“ karkaso suteiktomis galimybėmis lengvai manipuliuoti atvaizduojamais komponentais iš serverio pusės bei lanksčia duomenų bazės objektų

struktūra. Tai leido sukurti dinamišką paslaugų formų kūrimo mechanizmą, kuriuo sistemos administratorius gali sukurti paslaugų formas neturėdamas programavimo žinių.

Kiekvienos paslaugos duomenys rašomi į vieną rakto ir reikšmės struktūros objektą ir visa tai saugoma į viename duomenų bazės laukelyje. Šie duomenys, pasinaudojant JMS technologiją sistemų integracijai siunčiami išorinei sistemai apdoroti. Kadangi portalas epolicija.lt yra tik paslaugų duomenų priėmimo lygmuo, sukurti ir įtraukti naujas paslaugas yra labai paprasta ir greita.

Be paslaugų valdymo komponento, buvo realizuota ir naudotojo sąsaja, kuri glaudžiai susijusi su teikiamomis paslaugomis. Prisijungęs vartotojas gali peržiūrėti savo užsakytų paslaugų duomenis bei statusą, gauti pranešimus jam tinkamu būdu. Nors visos paslaugos kuriamos dinamiškai, naudotojas visą informaciją apie jas gali peržiūrėti vienoje vietoje.

Realizuota naudotojo sąsaja yra viešai pasiekiama adresu epolicija.lt (žr. 2 priedą). Sukurtas formų valdymo mechanizmas įgalino be sistemos atnaujinimo sukurti ir pradėti tiekti elektronines policijos paslaugas ir turi potencialą būti panaudotas kitose sistemose, kur reikalingas paslaugų valdymas ir naudotojo sąsajos kūrimas šioms paslaugoms.

INFORMACIJOS ŠALTINIŲ IR LITERATŪROS SĄRAŠAS

1. Apache Tomcat. [Interaktyvus]. [Žiūrėta 2012.11.25]. Prieiga per internetą: http://tomcat.apache.org/
2. Chen, H.; Cheng, R. ZK – Ajax without the Javascript Framework, 2007.
3. Chen, R.; Nair, S. Liferay Beginner's Guide, 2011.
4. Computer Literacy Tests: Are You Human? [Interaktyvus]. [Žiūrėta 2012.12.05]. Prieiga per internetą: http://www.time.com/time/magazine/article/0,9171,1812084,00.html
5. Dashorst, M.; Hillenius, E. "Wicket In Action" 2008.
6. Google Maps Javascript API V3 Reference. [Interaktyvus]. [Žiūrėta 2012.12.01]. Prieiga per internetą: https://developers.google.com/maps/documentation/javascript/reference
7. Minter, D.; Linwood, J. Building Portals with the Java Portlet API, 2004.
8. Nash, M. Java Frameworks and Components – Accelerate Your Web Application Development, 2003.
9. Liferay.com: Enterprise open source portal and collaboration software. [Interaktyvus]. [Žiūrėta 2012.10.01]. Prieiga per internetą: http://www.liferay.com/
10. Policijos departamentas prie Lietuvos Respublikos vidaus reikalų ministerijos. Viešosios policijos paslaugos „Pranešimų policijai teikimas" perkėlimo į elektroninę erdvę programinės įrangos sukūrimo paslaugų pirkimas, 2009.
11. Sezov, R. Liferay Administrator's Guide, 2nd Edition, 2008.
12. Svennerberg, G. Beginning Google Maps API 3, 2010.
13. The Open Group Base Specifications Issue 6, IEEE Std 1003.1, 2004 Edition. The Open Group. 2004. [Interaktyvus]. [Žiūrėta 2012.12.07]. Prieiga per internetą: http://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap09.html
14. TIOBE Programming Community Index. [Interaktyvus]. [Žiūrėta 2012.12.07]. Prieiga per internetą: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html
15. Vaynberg, I. Apache Wicket Cookbook, 2011.
16. ZK: Leading Enterprise Java Web Framework. [Interaktyvus]. [Žiūrėta 2012.10.15]. Prieiga per internetą: http://www.zkoss.org/

PRIEDAI

16-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“ teminė konferencija
Proceedings of the 16th Conference for Junior Researchers „Science – Future of Lithuania“

INFORMATIKA.INFORMACINĖS TECHNOLOGIJOS / INFORMATICS. INFORMATION TECHNOLOGIES

2013 m. Balandžio 9 d. Vilnius
09 April, Vilnius, Lithuania

eISSN 2029-7149
Article Number: *infotech.0002*

Informacinės Technologijos
Information Technologies

http://jmk.infotech.vgtu.lt

MODERNIŲ INTERNETINIŲ APLIKACIJŲ KŪRIMAS NAUDOJANT JAVA IR ZK INTERNETINĮ KARKASĄ

Nerijus Bučinskis
VGTU FMF Informacinių technologijų katedra
El. p. nerijus.bucinskis@gmail.com

Anotacija. Siekiant užtikrinti vartotojų aprūpinimą elektroninėmis paslaugomis, kompanijos vis dažniau renkasi ne fragmentuotus sprendimus, tokius kaip stacionarių programų išleidimas, bet lokalizuotus - internetines aplikacijas. Gaunamas mažesnis lankstumas, tačiau internetinė aplikacija yra daugiau nei internetinis puslapis. Pasirinkus šį sprendimą, gaunami reikšmingi privalumai: vienodai pasiekiamas didžiulis vartotojų skaičius, užtikrinama, jog elektroninės paslaugos tiekiamos unifikuotai ir patogiai. Be šių privalumų gaunami trūkumai - reikalingos plačios technologinės žinios tinkamai sukurti šiuolaikišką internetinę aplikaciją, ilgas kūrimo procesas. Šias problemas siekia išspręsti internetiniai karkasai, kurie palengvina ir suteikia daugiau galimybių kurti patikimas ir vartotojui draugiškas internetines aplikacijas. Šioje srityje ZK karkasas yra vienas iš lyderių.

Reikšminiai žodžiai: ZK, Java, Ajax, internetinis karkasas, šablonas, aplikacija, internetas, MVC, MVVM

1. Įvadas

Sparti verslo plėtra į elektroninę erdvę kelia vis didesnį poreikį kokybiškiems ir moderniems sprendimams, kurie leistų vykdyti elektroninę veiklą internete. Vien paprasto informacijos pateikimo nebeužtenka, reikalinga leisti vartotojui informaciją įvesti, ją keisti ir dalintis.
Pasiekti internetines aplikacijas labai patogu iš vartotojo pusės, tačiau jos dažnai turi ilgą, iteracinį kūrimo ciklą bei reikalauja daug technologinių žinių. Grafinei pusei naudojamos HTML, CSS, JavaScript technologijos (angl. *Front-End*), verslo logikai pasirenkama viena iš programinių kalbų, tokių kaip Java, C# ir kt. (angl. *Back-End*).
Java, kuriama ir prižiūrima Oracle Corporation, 2012 metų duomenimis buvo pati populiariausia programavimo kalba pasaulyje. Ji plačiai paplitusi kaip Android platformos pagrindas, tačiau geriausiai žinoma ir naudojama kliento-serverio internetinėms aplikacijoms kurti. Renkantis Java pagrindu paremtas technologijas, gaunami šie privalumai:

- **Atviro kodo bibliotekos**. Java turi plačią programuotojų bendruomenę, 2012 metais siekusią 10 mln. vartotojų, kuri dalijasi žiniomis, patarimais bei laisvai naudojamais kodo moduliais (Qian 2007).
- **Greitas ir efektyvus produkto kūrimo ciklas**. Dėl paprastos kodo sintaksės ir daugybės pasiekiamų pagalbinių modulių, produktas greitai pasiekia bandomąjį etapą.
- **Platus Web karkasų pasirinkimas**. Jų naudojimas leidžia supaprastinti aplikacijos kūrimą, ypač susijusį su grafine vartotojo sąsaja.

Karkasai suteikia unikalią galimybę gauti didelę dalį funkcionalumo, kuris būtinas sklandžiam aplikacijos veikimui, tačiau jo nereikia papildomai kurti nuo pagrindų. Rinkoje egzistuoja dešimtys šių karkasų, besiskiriančių naudojimo paprastumu, bendruomenės dydžiu, palaikymu ir t.t.

2. Internetiniai karkasai

Internetiniai karkasai (angl. *Web Frameworks*) yra nepakeičiama internetinių aplikacijų kūrimo dalis, kuri leidžia pradėti projektą nuo aukštesnio lygio realizacijos ir, priklausomai nuo tipo, nesirūpinti žemo lygio procesais, tokiais kaip duomenų siuntimas iš serverio į kliento aplinką ir pan. Internetiniai karkasai klasifikuojami į tris pagrindinius tipus:

- **Užklausos/veiksmo karkasai** (pavyzdžiai: Spring MVC, Struts 2 ir t.t.)
- **Serverio pusės karkasai** (pavyzdžiai: BlazeDS, Jax-RS ir t.t.)
- **Komponentiniai karkasai** (pavyzdžiai: GWT, JSF, ZK ir t.t)

Užklausos/veiksmo karkasai yra grindžiami MVC dizaino šablonu, ir skiriasi nuo kitų karkasų, jog neturi komponento sąvokos (priešingai nei komponentiniai karkasai). Juose turime galimybę keletą kartų naudoti tuos pačius veiklos kontrolerius bei kitus elementus, todėl

© Nerijus Bučinskis, 2013

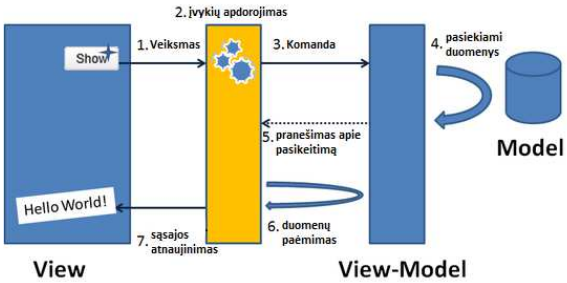
yra patikimi ir lengvai palaikomi. Dėl savo paprastumo populiaresni kuriant mažus projektus.
Serverio pusės karkasai orientuoti į verslo logiką ir dažniausiai skirti servisų kūrimui. Keletas realizacijų palaiko integraciją su grafine vartotojo sąsaja, tačiau yra skirti tik vidinei (servisų, procedūrų) struktūrai kurti.
Komponentiniai karkasai yra populiariausi, kadangi orientuoti į komponentus ir sukuria stacionarios (ne internetinės) aplikacijos įvaizdį. Tokia aplikacija būna pasiekiama per internetą (arba vidinį tinklą) todėl vartojai gali gauti vienodą turinį, paslaugas, nepriklausant nuo naudojamo įrenginio bei nedarant jokių pakeitimų juose (tokių kaip instaliacija ar konfigūracija). Komponentiniai karkasai pritaikyti aplikacijoms, kuriose reikalinga duomenų įvestis (formos). Jie tarpusavyje skiriasi pateikiama komponentų palete bei jų manipuliacijos įpatybėmis. Nors yra sudėtingesni, tačiau suteikia galimybę sukurti pažangias aplikacijas.
Vienas iš populiarių komponentinių karkasų yra ZK, parašytas Java programavimo kalba. Pirmoji versija buvo išleista 2006 m. vasario 27 dieną, ir per 6 metus, pasiekė 6-ąją versiją, kuri buvo išleista 2012 m. vasario 14 dieną. Per visus šiuos metus, komponentų kiekis, pasiekiamas programuotojams padidėjo dvigubai, nuo 60 iki 120, patobulintas objektų reikšmių valdymo mechanizmas, sukurtas patogus API, kuriuo programuotojui suteikiama laisvė ne tik naudoti, bet ir keisti standartinį funkcionalumą. Pagrindiniai ZK privalumai:

- Per 120 Ajax pagrindu veikiančių komponentų. Komponentų paletėje randame lango, medžio, puslapiavimo ir kt. komponentus, taip pat įvairius internetinio puslapio struktūrai formuoti skirtus elementus.
- *Drag-and-drop* palaikymas.
- Automatinis reikšmių užpildymas (angl. *Autocomplete*)
- Uždelstas komponentų užkrovimas.
- Įvykių pagrindu (angl. *Event-Driven*) su serveriu susietas modelis.
- Nereikia papildomai valdyti bendravimo tarp serverio ir kliento, visi veiksmai dedami į eilę ir sinchroniškai apdorojami.

Puslapių kūrimas beveik visiškai eliminuoja būtinybę žinoti CSS, HTML ar JavaScript, kadangi puslapio formavimas vyksta aprašant ZUL failus, XML pagrindu sukurta ZUML kalba (angl. *ZK User Interface Markup Language*). Tokiu būdu programuotojui užtenka aprašyti komponentą, suteikti jam identifikatorių, o vidinė logika, apdorojusi ZUL failą, paverčia jį į vartotojui rodomą HTML failą (Schumacher 2008).

3. MVVM struktūra

Aplikacijos infrastruktūra yra kertinis akmuo jos kūrime. Tinkamai realizuota architektūra leidžia pasiekti maksimalų efektyvumą ir našumą. Apgalvota architektūra užtikrina lengvai prižiūrimą ir tobulinamą visos aplikacijos programinį kodą.
ZK karkase didelis dėmesys skiriamas duomenų valdymui tarp serverio bei duomenų atvaizduojamų grafinėje sąsajoje, todėl stengiamasi kuo labiau atskirti duomenis ir verslo logiką nuo vaizdavimo lygmens. Tai glaudžiai susiję su naudojamu MVVM projektavimo šablonu.
MVVM (angl. *Model/View/View-Model*) architektūrinis šablonas, pristatytas Martino Fowlerio, yra vienas iš MVC (angl. *Model/View/Controller*) atmainų. Jis pasižymi papildomu duomenų logikos atskyrimu nuo vaizdavimo sluoksnio. Tai pasiekiama įvedant modelio vaizdo (angl. V*iew-Model*) sąvoką, kuri tampa tarpiniu sluoksniu tarp duomenų ir vaizdavimo sluoksnio. Kaip šis šablonas realizuojamas ZK karkase (žr. 1 pav.):



1 pav. Duomenų valdymo infrastruktūra: 1) Veiksmas - vartotojo inicijuotas veiksmas iš interneto naršyklės su ZK kurtame puslapyje; 2) Įvykių apdorojimo variklis; 3) Komanda - užklausa pranešanti duomenų poreikį; 4) Modelio vaizdo lygmuo - duomenų paėmimas iš tikrosios saugyklos; 5) Pranešimas apie duomenų pasikeitimą; 6) Duomenų paėmimas per modelio vaizdo sluoksnį; 7) Vartotojo grafinės sąsajos atnaujinimas su naujais duomenimis.

Fig. 1. Data management infrastructure: 1) Action - user initiated action from browser in web application created with ZK; 2) Event handling engine (binder); 3) Command - request for data; 4) View-Model layer - getting data from actual data store; 5) Notification about data change; 6) Acquiring data through model-view layer; 7) Updating graphical user interface with new data.

Ši variacija, lyginant su MVC, taip pat turi tris sluoksnius (lygmenis) kur View ir Model atlieka analogišką vaidmenį. Skirtumas tas, jog MVVM šablone, View-Model yra specialios veiklos kontroleris View sluoksniui. Jis atsakingas už duomenų pateikimą iš modelio View sluoksniui bei reikalingą logiką skirtą užtikrinti užklausų valdymui (Osmani 2012).
ViewModel sluoksnis savyje apima View būseną ir veiklą, bet išlaiko atstumą nuo fizinės grafinės dalies (neturi jokių žinių apie grafinėje vartotojo sąsajoje rodomus komponentus). Taip ViewModel išlaiko pagrindines MVVM šablono charakteristikas, užtikindama mažą susietumą tarp sluoksnių ir paprastą struktūrą.
Raktinė savybė būdinga MVVM šablonui yra ta, jog ModelView lygmuo nieko nežino apie View sluoksnio vizualinius komponentus, o tai garantuoja vienos krypties priklausomybę nuo View lygmens. Tai išvengia nepageidaujamų efektų, kuomet glaudžiai siejamas prezentacinis ir modelio lygmenys, ir suteikia tokius privalumus:

1. **pritaikomas dizainas-pagal-kontraktą**: kol išlaikomas kontraktas (kokius duomenis rodyti, kokius veiksmus atlikti), grafinės dalies ir VieModel lygmenų programavimas gali vykti paraleliai ar netgi atskirai. Abi pusės neįtakos kitos progreso.
2. **Laisvas susiejimas su vaizdavimo lygmeniu**: grafinė vartotojo sąsaja gali būti keičiama daugelį kartų, neįtakojant žemiau esančių veiklos ir informacijos sluoksnių.
3. **Geresnis perpanaudojamumas**: naudojant bendrą ViewModel sluoksnį, jam galima pritaikyti skirtingus View sluoksnius. Tai ypač naudinga kai reikia atvaizduoti daugiau duomenų ant didesnio ekrano, ar analogiškai - mažiau ant mažo ekrano.
4. **Geresnis testavimas**: kadangi vaizdinis sluoksnis nėra surištas su veiklos ar modelio sluoksniais, galima parašyti testus nenaudojant grafinių elementų.

4. ZUL failai

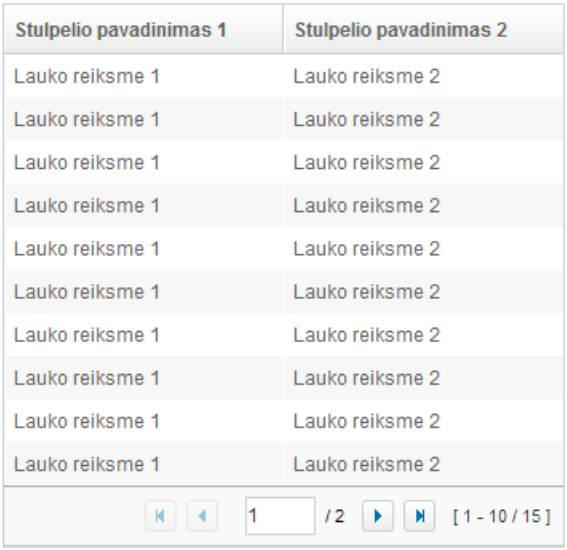
Nors ZK yra internetinis karkasas, juo naudojantis beveik nereikia rašyti HTML elementų, vietoje to naudojami XML struktūros ZUL failai.
Atvaizduojant informaciją internete, dažnai tenka perteikti objektų rinkinį į ekraną. Šis konkretus atvejis realizuojamas parašius keletą ZUL eilučių ZUML kalba ir aprašius objektą, valdantį duomenų perdavimą. Toliau pateikiamas tokio funkcionalumo kodo fragmentas (žr. 2 pav.):

```
<listbox id="lentele" model="@load(vm.model)">
    <listhead>
        <listheader label="Stulpelio pavadinimas 1" />
        <listheader label="Stulpelio pavadinimas 2" />
    </listhead>
    <template name="model">
        <listitem>
            <listcell label="Lauko reiksme 1" />
            <listcell label="Lauko reiksme 2" />
        </listitem>
    </template>
</listbox>
```

2 pav. ZUL failo fragmentas. Čia: *listbox* - reikšmių atvaizdavimo komponentas, rodantis visas reikšmes lentelės pavidalu; *listhead* ir *listheader* - aprašo lentelės stulpelius; *template* - indikatorius jog sekantis kodas yra kartojamas tiek kartų, kiek yra reikšmių modelyje; *listitem* ir *listcell* - aprašo vienos eilutės atvaizdavimą; *vm* - modelio vaizdo objektas, valdantis duomenų perdavimą; *model* - reikšmių rinkinys.

Fig. 2. ZUL file fragment. Here: *listbox* - component displaying all model values in table-like component; *listhead* and *listheader* - tells how to create table header; *template* - indicating that following code will be duplicated for all values in model; *listitem* and *listcell* - tells how to display single row; *vm* - view-model object, handling data, *model* - collection of objects.

Šis aprašas paverčiamas į sudėtingą HTML ir JavaScript struktūrą, kuri vartotojo interneto naršyklėje pateikia duomenis interaktyvios lentelės pavidalu, su galimybe valdyti šį komponentą programiškai. Komponentas, JavaScript pagalba tampa visiškai interaktyvus, t.y. galima pažymėti kurį nors elementą kaip pasirinktą arba gauti informaciją apie vartotojo inicijuotą pasirinkimą, taip pat kitus įvykius susijusius su šiuo komponentu (Chen 2007).
Papildomai pridėjus *mold='paging' pageSize='10'* prie *listbox*, ZK grafinės sąsajos generavimo variklis pridedą automatinį puslapiavimą, kai atvaizduojamų objektų skaičius pasiekia daugiau nei 10 vienetų. Gauname galimybę eiti pirmyn, atgal per puslapius bei pasirinkti tikslų puslapį. Nedideli kodo pasikeitimai sukuria sudėtingą funkcionalumą, kurio nereikia realizuoti programuotojui (žr. 3 pav.).
listbox yra tik vienas iš daugelio ZK komponentų, kuriais aplikacijos kūrėjas gali naudotis, juos tobulinti, keisti, sujunkti. Be konkrečiuos paskirties komponentų, dar gauname eilę puslapio struktūros komponentų.



3 pav. *listbox* komponentas su puslapiavimo elementu ir duomenimis vartotojo interneto naršyklėje.

Fig. 3. *listbox* component with paging element and data in user web browser.

5. Integracija su kitais programiniais paketais

Sukurti visapusiškai funkcionalią, saugią ir vartotojui malonią aplikaciją vien internetinio karkaso nepakanka, jai reikalingi papildomi programiniai paketai. Dėl šios priežasties, remiantis karkasiniu pagrindu, svarbu turėti galimybę prijungti papildomus modulius ir jais naudotis.
Moderniai internetinei aplikacijai, būtina užtikrinti saugumą ir sąsaja su duomenų baze. Tai realizuota atviro kodo programiniuose paketuose, tokiuose kaip Spring ir Hibernate. Spring Framework yra labai populiarus tarp kompanijų, kuriančių internetinių aplikacijų sistemas. Šioms kodo bibliotekoms integruoti ZK turi vidinį mechanizmą, skirtą prijungti šių paketų teikiamas funkcijas.

Turint aplikaciją, parašytą naudojantis JSP arba Struts karkasais, ZK suteikia galimybę nedarant didelių pokyčių, permigruoti visą kodą ant ZK. Dėl šio palaikymo, ZUL faile galima aprašyti kodo fragmentus iš kitų internetinių karkasų, kur jie nepastebimai integruojasi su esančiais ZK komponentais.
Dažnai aplikacijų kūrėjai renkasi populiarius portalų struktūros programinius paketus, tokius kaip Liferay, kuriuose jau būna paruošta visa turinio ir grafinės sąsajos valdymo infrastruktūra. ZK turi palaikymą su tokiais paketais, todėl galima kurti portalo įskiepius (angl. P*ortlet*), kurie vėliau talpinami į portalo serverį.

6. Pasiekiamumo įvairovė

Šiuolaikinis vartotojas gali prisijungti prie interneto ir naudotis paslaugomis plačiu spektru elektroninių įrenginių. Naudotis elektroninėmis paslaugomis internete galima net tik stacionariais kompiuteriais, bet ir mobiliaisiais telefonais, planšetiniais kompiuteriais, todėl svarbu, kad internetinės aplikacijos optimaliai veiktų tarp skirtingų įrenginių ir platformų.
ZK optimizuoja komponentus taip, kad jie atrodytų vienodai skirtingose aplinkose, ar tai būtų pele ir klaviatūra valdomas įrenginys, ar įrenginys su lietimui jautriu ekranu. Taip programuotojas gali sutaupyti laiko bandydamas pasiekti aplikacijos būseną, kada ji vienodai atrodo tarp skirtingų įrenginių (bei turi tą patį turtingą funkcionalumą). Sutaupytas laikas gali būti skiriamas užtikrinti teisingą vidinės logikos veikimą.
Realizacija grindžiama naujomis CSS 3 funkcijomis, kurios suteikia galimybę gauti įrenginio, kuriuo atidaroma internetinė aplikacija, ekrano aukštį ir plotį. Žinant šiuos parametrus galime optimizuoti vaizduojamą turinį pagal ekrano rezoliuciją.

7. Išvados

Pasirinkus Java programavimo kalbą, gauname visavertį sprendimą internetinių aplikacijų kūrimui. ZK, pažangus internetinis šablonas, parašytas Java kalba, suteikia galimybę kurti modernias, šiuolaikiškas ir vartotojui draugiškas internetines aplikacijas. Be visų privalumų gaunamų iš vartojamosios pusės, kurti su ZK karkasų yra paprasta, patogu ir lengva. Netgi neturint pakankamai HTML, JavaScript ar CSS žinių, galima parašyti funkcionalią ir kokybišką internetinę aplikaciją.
ZK internetiniame karkase sudarytos galimybės naudoti populiarius programinius paketus, tokius kaip Spring, Hibernate bei įterpti fragmentus iš Struts ar JSP, sklandžiai papildant komponentų paletę.

Literatūra

Qian K. 2007. Java Web development illuminated.

Osmani A. 2012. Learning JavaScript Design Patterns: 125–138 p.

Chen H. Cheng R. 2007. ZK – Ajax without the Javascript Framework.

Schumacher H. Stauble M. 2008. ZK Developer's Guide.

DEVELOPING MODERN INTERNET APPLICATIONS USING JAVA AND ZK WEB FRAMEWORK

N. Bučinskis

Summary

Developing rich and reliable web applications is made easy with Java and ZK web framework. Component based framework provides not only great architecture, but also a wide selection of usable components. MVVM architecture ensure loose coupling between layers, which consequently brings better maintainability. The amount of given components is more than sufficient to create simple web applications and is a great start for creating big enterprise applications. The support for third party modules and libraries with ability to adapt to various platforms and devices, ZK web framework is one of the best tools for creating modern web applications.

Keywords: ZK, Java, Ajax, web framework, pattern, application, internet, MVC, MVVM.

2 priedas.

Vilniaus Gedimino Technikos universitetui

Informacinių technologijų katedrai

Pažymime, kad UAB „INSOFT“ programuotojo Nerijaus Bučinskio pasirinkta baigiamojo bakalauro darbo tema „Elektroninių paslaugų naudotojo sąsajos kūrimas epolicija.lt portale“ yra suderinta su projekto vadovais, yra aktuali mūsų įmonei ir tiesiogiai susijusi su jo atliekamais darbais. Sistema įdiegta ir naudojama didelėje valstybės biudžetinėje įstaigoje.

Projektų vadovas,

Povilas Bertašius

UAB „Insoft“
Ozo g. 10A
LT-08200 Vilnius
Lietuva

Tel.: +370 5 210 0660
Faks. +370 5 243 2131
El. paštas: info@insoft.lt
www.insoft.lt

Juridinio asmens kodas: 302294870
PVM mokėtojo kodas: LT100004466518
Bankas: AB SEB bankas
A.s. Nr. LT48 7044 0600 0665 7896