ŠIAULIŲ UNIVERSITETAS

TECHNOLOGIJOS, FIZINIŲ IR BIOMEDICINOS MOKSLŲ FAKULTETAS

INFORMATIKOS IR MATEMATIKOS KATEDRA

Aivaras Karnatka

Informatikos specialybės II kurso nuolatinių studijų studentas

Buitinių paslaugų tarpininkavimo sistemos modelio tyrimas

Domestic service mediation system model study

MAGISTRO DARBAS

Darbo vadovas:
Doc.dr. L. Kaklauskas

Recenzentas:

Šiauliai, 2017

„Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota, o paruošta naudojant literatūros sąraše pateiktus informacinius šaltinius bei savo tyrimų duomenis.“

Darbo autoriaus _____
(vardas, pavardė, parašas)

DARBO TIKSLAS IR UŽDAVINIAI

Darbo tikslas: suprojektuoti buitinių paslaugų tarpininkavimo sistemos modelį ir jį ištirti.

Darbo uždaviniai:

1. Išanalizuoti ir įvertinti virtualių tarpininkavimo sistemų funkcionalumą;
2. Technologinių sprendimų, taikomų tarpininkavimo sistemose, analizė;
3. Suprojektuoti tarpininkavimo sistemos modelį;
4. Realizuoti tarpininkavimo sistemos prototipą, jį ištestuoti ir išanalizuoti gautus rezultatus.

Darbo vadovas _____
(vardas, pavardė, parašas)

TERMINŲ ŽODYNAS

SQL – kalba, skirta aprašyti duomenis ir manipuliuoti jais reliacinių duomenų bazių valdymo sistemose.

URL – (Universalus informacijos šaltinio adresas, angl. Uniform Resource Locator) būdas aprašyti tam tikro informacinio resurso (tinklalapio, serveryje patalpinto failo) adresą internete.

BPMN – verslo procesų modeliavimo standartas.

PYPL – programavimo kalbų populiarumo apžvalga, sukurta remiantis Google sistemoje užduodamų programavimo kalbų pamokų užklausų rezultatais.

Git – viena iš populiariausių versijų kontrolės sistemų.

GitHub – populiariausia debesų talpykla git'o saugykloms.

MVC – principas, kuriuo paremti veikia daugelis populiarių programavimo karkasų.

TURINYS

ĮVADAS

Išsivysčiusiose šalyse industrinę visuomenę keičia informacinė. Lietuvoje, kaip ir daugelyje kitų Europos ir pasaulio šalių, sparčiai auga informacinių technologijų ir telekomunikacijų naudojimas, įsisavinamos vis modernesnės technologijos. Vienos iš telekomunikacijų bendrovės užsakymu atliktas rinkos tyrimas parodė, kad Lietuvoje, Latvijoje ir Estijoje naudingas ir praktiškas paslaugas savo išmaniuosiuose įrenginiuose naudoja 54–56 proc. vartotojų (baltic–amadeus.lt, 2016). Išmaniųjų technologijų populiarumo augimas parodo, kad žmonės supranta ir išmoksta „įdarbinti" naująsias technologijas savo kasdienio gyvenimo problemoms spręsti.

Informacinėms technologijoms vis intensyviau skverbiantis į kasdienį žmonių gyvenimą, vis sparčiau auga ir poreikis kurti taikomąsias programas, skirtas kasdieniams darbams palengvinti ar atlikti įvairias užduotis, pavyzdžiui, išsikviesti taksi. Vienas iš pavyzdžių – 2011 metais savo veiklą pradėjusi transporto tarpininkavimo paslaugas teikianti programa „Uber''. Ši programa per penkerius metus tapo viena populiariausių ir brangiausių programų pasaulyje, pritraukiančia per milijardą vartotojų (uber.com, 2016). Lietuviškas šios sistemos pavyzdys galėtų būti programa „e–taksi", kuri taip pat yra tarpininkas tarp keleivio ir vežėjo, tačiau vežimo paslaugą teikia licenciją turinčios transporto įmonės, o ne bet kuris vairuotojas (verslas.in, 2016).

Šias laikais, kai elektroninė prekyba įgauna vis didesnį pagreitį, daugelis žmonių kasdienes prekes įsigyja neišėję iš namų – tiesiog internete. Viena iš populiariausių, didžiausių ir labiausiai lankomų pasaulyje internetinių parduotuvių yra „eBay''. Kaip bebūtų keista, pati parduotuvė neturi nė vienos savo prekės, tai lyg didelė tarpininkavimo sistema tarp pardavėjo ir pirkėjo.

Tarpininkavimo sistemos šiais laikais tampa vis aktualesnės, jų poreikis vis auga. Šio tipo sistemų poreikio augimui įtakos turėjo išmaniųjų įrenginių atsiradimas. Internetinėmis programomis galime naudotis visur ir visada, problemas galime spręsti tiesiog eidami gatvėje – šis faktorius vartotoją vis labiau įtraukią į kibernetinį pasaulį, o jo kūrėjui suteikia vis didesnį stimulą jį plėsti, kurti naujas galimybes, programas.

Buitinių paslaugų srityje tarpininkavimo sistemos taip pat naudingos. Jos padeda užsakovams nepasimesti skelbimų gausoje, ieškant tinkamo meistro, auklės, valytojo,

kitas buitines paslaugas teikiančio asmens ar įmonės. Deja, šiandien Lietuvoje tokio pobūdžio nemokamų tarpininkavimo sistemų, kurios būtų patogios įvairiems vartotojų tipams, rastų tik tinkamus ir patikimiausius meistrus, nėra.

Darbo tikslas: suprojektuoti buitinių paslaugų tarpininkavimo sistemos modelį ir jį ištirti.

Darbo uždaviniai:

1. Išanalizuoti ir įvertinti virtualių tarpininkavimo sistemų funkcionalumą;
2. Technologinių sprendimų, taikomų tarpininkavimo sistemose, analizė;
3. Suprojektuoti tarpininkavimo sistemos modelį;
4. Realizuoti tarpininkavimo sistemos prototipą, jį ištestuoti ir išanalizuoti gautus rezultatus.

1. TARPININKAVIMO SISTEMOS

1.1 Tarpininkavimo sistemos samprata

Tarpininko terminu gali būti įvardijamas rinkos dalyvis, kuris parduoda prekę, tačiau jos nesukuria ir nevartoja (Gatautis, Neverauskas, Snieška, 2002). Pagal pateikiamus apibrėžimus, tarpininkas nusakomas kaip „Rinkos dalyvis, esantis tarp tiekėjo ir vartotojo. Tarpininkas veikia kaip ekonominis agentas, kuris perka iš tiekėjų, o po to perparduoda pirkėjams ar padeda pirkėjams ir pardavėjams susitikti ir sudaryti sandorį“ (Dzemydiene, Okulič–Kazarinas, 2016).

Užsakomųjų paslaugų vykdymą pasitelkiant virtualaus tarpininkavimo procesą galima apibūdinti kaip internetinėje erdvėje susiformavusį ryšį ar santykius tarp reikalingų paslaugų ieškančių bei jas teikiančių ir siūlančių subjektų (Vasicek, 2005). Visus subjektus, kurie organizuoja apsikeitimą informacija apie prekes ir paslaugas, naudodamiesi elektroniniais kanalais, galima vadinti virtualiaisiais tarpininkais (Gupta, Woodside, 2006).

1.2 Virtualių tarpininkavimo sistemų tipai

Virtualiais tarpininkais galima įvardyti ,,kažką” nepriklausomą, administruojantį intelektualų tarpininkavimo sprendimą, sukuriantį pridėtinę vertę tiek pirkėjui, tiek tiekėjui, atliekant sandorį virtualioje rinkoje (Fairchild, 2004).

Virtualiųjų tarpininkų teikiamas e. paslaugas galima klasifikuoti pagal:

- naudojimą atskiruose sandorio įgyvendinimo etapuose (informacijos paieškos, derybų, kokybės įvertinimo, nuosavybės teisių apsaugos);
- veiklos organizavimo etapus (informacijos paieška, sandorio koordinavimas, sandorio kontrolė);
- pasirinktų sandorio veiklų tobulinimą (finansinės informacijos integravimas, klientų užsakymų informacijos integravimas, gamybinių procesų gerinimas ir standartizavimas, laikomų atsargų mažinimas, personalo valdymo standartizavimas);
- taikomas technologijas (pvz., programavimo architektūros pasirinkimą)

Norint pateikti e. tarpininkų veiklos vystimosi scenarijus (Giaglis, Klein, 2010), galima juos grupuoti pagal kuriamą pridėtinę vertę:

- Dis–tarpininkai (angl. Disintermediares), savo veiklą apribojantys siaurąja tarpininkavimo veiklos samprata, informacijos tarp ieškančiųjų bei siūlančiųjų paskleidimu.
- Re–tarpininkai (angl. Reintermediares), savo teikiamas paslaugas pirmiausia orientuojantys į tiekėjų poreikių tenkinimą. Tiekėjų pasiūlymo pateikimas rinkai originaliais Re–tarpininko kanalais kuria pridėtinę vertę tiekėjui per informacijos pateikimo efektyvumą (mažina sandorio kaštus), papildomas paslaugas (vartotojų instruktavimą, apmokymą, atgalinio ryšio palaikymą), koncentracija į nišines rinkas;
- Kibernetiniai tarpininkai (angl. Cybermediares), rinkoje atliekantys infrastruktūros dalies vaidmenį. Kibernetiniai tarpininkai kuria pridėtinę vertę sandorio e. erdvėje dalyviams, jiems asistuodami, atlikdami dalį rutininių operacijų (apmokėjimo, produkto išsiuntimo, virtualaus eksponavimo ir pan.)

1.3 Esamų tarpininkavimo sistemų apžvalga

Lietuvoje ir pasaulyje yra sukurta daug įvairaus pobūdžio internetinių tarpininkavimo sistemų. Šiame skyriuje apžvelgsime keletą tarpininkavimo sistemų, aptarsime jų funkcionalumą, naudojimo sąlygas ir t.t.

Betreut.de (Etalinq, 2013) – tarpininkavimo sistema, suvedanti vartotoją su priežiūros, slaugos ir namų ūkio paslaugų teikėjais. Betreut.de padeda rasti įvairiausias paslaugas teikiančius žmones netoli savo gyvenamosios vietos. Betreut.de siūlo patogią interneto svetainę, kurioje kiekvienas norintis gali greitai rasti tinkamiausią paslaugų teikėją. Betreut.de padės susirasti reikiamą paslaugą ne tik didžiuosiuose Vokietijos miestuose – Berlyne, Hamburge, Miunchene – bet ir Austrijoje bei Šveicarijoje. Betreut.de svetainėje gali užsiregistruoti tiek paslaugų teikėjai, susikurdami profilį ir jame pateikdami savo nuotrauką, įvairiausią informaciją apie savo kvalifikaciją bei teikiamas paslaugas, tiek paslaugų, jų teikėjų ieškantys žmonės. Registracija svetainėje nemokama. Paslaugų ieškantis klientas, svetainėje įvedęs savo pašto indeksą mato arčiausiai jo gyvenamosios vietos esančius paslaugų teikėjus, jis taip pat gali pasirinkti,

kokius skelbimus norėtų peržiūrėti bei užsisakyti jam įdomių skelbimų naujienlaiškį. Naudojimasis šia sistema yra nemokamas, kaip ir registracija svetainėje. Klientas sumoka šios svetainės savininkams tik tada, kai susisiekia su paslaugos teikėju. Paslaugų teikėjams registracija ir savo profilio sukūrimas bei kreipimasis į potencialius klientus taip pat nekainuoja. Tačiau jei paslaugų teikėjas nori patekti į matomiausią puslapio vietą, aukščiau pakilti paslaugų teikėjų sąraše, už tai jau turi susimokėti. Kartu su tarpininkavimo paslauga betreut.de siūlo ir nemažai įvairių straipsnių slaugos, namų ūkio priežiūros ir kitomis aktualiomis temomis.

manopaskola.lt (spec.lt. 2016) – elektroninė paskolų tarpininkavimo sistema, padedanti žmonėms susirasti optimalų būsto paskolos finansavimo sprendimą. Šioje sistemoje užpildę vieną paraišką, žmonės gauna įvairių bankų paskolų pasiūlymus per kelias darbo dienas. Vienas iš pagrindinių šios sistemos privalumų – greitas ir patogus paskolos pasirinkimo bei gavimo procesas ir galimybė įvairiais aspektais palyginus bankų siūlomas paskolų sąlygas, išsirinkti geriausią variantą. Taip pat prie privalumų paminėti galima būtų patogią ir aiškią vartotojo sąsają. Prie sistemos prisijungęs žmogus iškart mato norimos paskolos formą, kurią užpildžius pateikiamas pasiūlymas. Kitaip tariant, vartotojui nereikia klaidžioti po sistemą.

Tarpininkavimo sistemos paslaugos klientams nieko nekainuoja.

raskgreitai.lt (Daiva projects, UAB, 2013) – Lietuvoje veikianti internetinė buitinių paslaugų tarpininkavimo sistema. Ji skirta tiek ieškantiems naujų klientų, tiek patikimų paslaugų teikėjų. Šios sistemos veikimo principas: klientui užpildžius meistro paieškos užklausos formą, sistema išsiunčia pasiūlymus visiems meistrams, kurie atitinka ieškomo meistro kategorija ir miestą. Tuomet užklausas gavę meistrai susisiekia su užsakovu. Klientai užklausas paslaugoms gali teikti nemokamai. Paslaugų teikėjai portalą išbandyti nemokamai gali 182 dienas. Vėliau yra galimybė rinktis, už kurias portalo paslaugas mokėti, atsižvelgiant į konkretaus verslo poreikius. Vienas iš pagrindinių sistemos privalų – aiškus paslaugų užsakymo proceso (veiksmų sekos) pateikimas.

Taskbob (taskbob.com, 2016) – tai 2014 metais Indijoje pradėjus veikti buitinių paslaugų užsakymo sistema. Šios sistemos veikimo principas: klientas pasirenka tą buitinių paslaugų kategoriją, kurios meistras jam reikalingas, įveda savo adresą, telefono numerį, ir sistema, surandusi jam tinkančius meistrus, išsiunčia jiems pranešimus. Taip

pat klientas gali tiesiogiai sistemoje susirašinėti su jam tikusiu meistru. Klientams, ieškantiems meistrų, ši buitinių paslaugų tarpininkavimo sistema yra nemokama, o paslaugas teikiantys meistrai, norėdami užsiregistruoti sistemoje, turi mokėti registracijos mokestį. Vienas iš pagrindinių sistemos privalumų yra tai, kad ji pilnai pritaikyta naudotis tiek išmaniaisiais įrenginiais, tik kompiuteriais.

Apžvelgus panašaus pobūdžio tarpininkavimo sistemas, galima išskirti tokius esminius požymius: veikimo logika, naudojimo sąlygos, informacijos pateikimas vartotojui, sistemų veikimo diapazonas. Žvelgiant į sistemas iš techninės pusės, reikėtų išskirti šiuos požymius: sukūrimo technologija, naudojimo platforma, sistemos navigacija, paslaugų užsakymo procesas.

Pirmoje lentelėje pateiktas aptartų tarpininkavimo sistemų palyginimas. Šio palyginimo tikslas – išsiaiškinti panašaus pobūdžio tarpininkavimo sistemų privalumus ir trūkumus.

1.3.1 Tarpininkavimo sistemų lyginamosios analizės kriterijai

Efektyvios internetinės svetainės ar sistemos samprata skamba labai skirtingai. Vieni straipsnių autoriai pasisako už vienus, kiti už kitus sistemų kūrimo aspektus. Efektyvi internetinė sistema yra ta, kurią vartotojai linkę dažnai naudoti (Li, Zhang, 2005).

Yra pateiktos rekomendacijos, kaip sukurti efektyvią internetinę sistemą ar svetainę. Tam tikslui įgyvendinti reikia suprojektuoti sistemą taip, kad ji būtų lengvai randama internete, sukurti paprastą ir aiškią vartotojui navigaciją, suteikti vartotojams priežastį ir galimybes naudotis sistema, padaryti sistemą vizualiai patrauklią (Cell, 2007). Remiantis šiais patarimais, buvo pasirinkti šie vertinimo kriterijai:

Naudojimo platforma, pritaikymas išmaniesiems įrenginiams – kaip vartotojai pasiekia sistemą ir ja naudojasi (per interneto naršyklę, mobilią programą – *angl. apps* – ar konsolinę programą). Jeigu sistema pasiekiama per interneto naršyklę, svarbu, ar ji prisitaiko prie įvairių įrenginių ekranų rezoliucijų.

Navigacija – kadangi dauguma internetinių sistemų ar svetainių turi vidinius puslapius, būtina suteikti vartotojams navigacinį palaikymą, kad jie turėtų galimybę

laisvai naršyti po sistemą. Geriausia navigacinė sistema leidžia lankytojui pasiekti bet kokį puslapį vienu spragtelėjimu (Nelson, 2000).

Naudojimo sąlygos – kokiomis sąlygomis vartotojai gali naudotis sistema (mokamai, nemokamai, dalinai mokamai). Jei sistema galima naudotis dalinai mokamai, svarbu, kokios suteikiamos naudojimosi galimybės.

Žemiau pateikti kriterijai pasirinkti atsižvelgiant į kuriamą sistemą, jos numatomas galimybes.

Veikimo logika – kaip vartotojas pradeda naudotis sistema, apžvelgiama sistemos sąveika su vartotoju.

Informacijos pateikimas – kaip pateikiama informacija apie sistemą, jos sąlygas, naudojimosi instrukcija.

Paslaugų užsakymas – apžvelgiamas lyginamų sistemų paslaugų užsakymo mechanizmas.

1.3.2 Tarpininkavimo sistemų lyginamoji analizė

1.Lentelė *Tarpininkavimo sistemų palyginimas.*

Kriterijus	**Betreut.de**	**manopaskola.lt**	**raskgreitai.lt**	**Taskbob**
Veikimo logika	Vartotojas, ieškantis paslaugos teikėjų pagal įvestą pašto indeksą, mato arčiausiai esančių paslaugų teikėjų informaciją.	Vartotojas, užpildęs paraišką, gauna įvairių bankų paskolų pasiūlymus	Klientas užpildo meistro paieškos užklausos formą, sistema išsiunčia pasiūlymus visiems meistrams, atitinkantiems ieškomo meistro kategoriją ir miestą, meistrai gavę užklausas susisiekia su užsakovu	Klientas pasirenka tą buitinių paslaugų kategoriją, kurios meistro jam reikia, įveda savo adresą, telefono numerį, tuomet sistema suranda jam tinkančius meistrus ir išsiuničia jiems pranešimus.
Naudojimo sąlygos	Dalinai nemokama	Nemokama	Mokama	Dalinai nemokama

Informacijos pateikimas	Informacija išdėstyta aiškiai, pradėti naudotis sistema galima esant tituliniame puslapyje.	Sistema pasiekiama esant tituliniame puslapyje, taip pat pateiktas sistemos aprašymas. Vartotojas gali pradėti naudotis sistema iškart, kai tik ją įsijungia.	Aiškiai išdėstytas sistemos veikimas, pradėti naudotis sistema gana paprasta, tačiau kituose žingsniuose informacija pateikiama klaidinančiai.	Informacijos pateikimas nėra aiškus, sistema daugiau orientuota į vartotoją, kuris planuoja ja naudoti išmaniajame įrenginyje.
Veikimo diapazonas	Vokietija, Austrija, Šveicarija	Lietuva	Lietuva	Indija
Sukūrimo technologija	Java	PHP	ASP.NET	Ruby (Ruby on Rails)
Naudojimo platforma	Interneto naršyklė	Interneto naršyklė	Interneto naršyklė	Sistema, veikianti per interneto naršyklę, taip pat yra sukurta aplikacija išmaniesiems įrenginiams.
Navigacija	Navigacija klaidinanti, meniu punktai ne visada veda ten pat, yra situacijų, kai jų nebelieka.	Navigacija aiški, visi puslapiai pasiekiami, esant bet kuriame kitame puslapyje.	Navigacija klaidinanti, meniu punktai ne visada veda ten pat, yra situacijų, kai iš sistemos išeinama į kitą puslapį.	Navigacija aiški, visi puslapiai pasiekiami, esant bet kuriame kitame puslapyje.
Paslaugų užsakymas	Sistema pateikia tinkamus paslaugų teikėjus, su jais vartotojas susisiekia pats.	Sistema vartotojui pateikia paskolų pasiūlymus, nėra numatyta, ką toliau vartotojas darys sistemoje.	Sistema vartotojui suranda tinkamus meistrus ir išsiunčia jiems pranešimus, meistrai skambina klientams ir tariasi su jais.	Sistema vartotojui suranda tinkamus meistrus ir išsiunčia jiems pranešimus, meistrai skambina klientams ir tariasi su jais.
Pritaikymas išmaniesiems įrenginiams	Nėra	Nėra	Nėra	Yra

Visos keturios lygintos sistemos yra skirtos spręsti buitinėms problemoms. Visos sistemos yra sukurtos naudojant internetui skirtas programavimo kalbas, tai reiškia, kad jas galima pasiekti per interneto naršyklę. TaskBob sistema turi mobilią aplikaciją, sukurtą Java programavimo kalba. Tai reiškia, kad vartotojas gali pasirinkti, kaip

patogiau naudotis sistema, o tai yra didelis privalumas. Tik išmaniesiems įrenginiams (mobili aplikacija) pritaikyta sistema, kurią naudos platus vartotojų ratas, nėra gerai. Tai iliustruoja ir UAB Bitė Lietuva atlikto tyrimo rezultatai: 60% išmaniųjų telefonų savininkų Lietuvoje savo įrenginiuose neturi mobiliojo interneto paslaugos, o 41 proc. juos naudoja tik skambučiams ir trumpųjų žinučių rašymui. Taigi, šiems vartotojams sistema būtų arba nepasiekiama, arba nenaudinga (jei nebūtų interneto).

Tik viena iš lygintų sistemų (Taskbob) yra pritaikyta išmaniesiems įrenginiams. Kad jiems nėra pritaikytos kitos – didelis minusas. Informacinės visuomenės plėtros komiteto prie Susisiekimo ministerijos užsakyto tyrimo duomenys rodo, kad jau 2013 m. IV ketv. 28,4 proc. Lietuvos gyventojų jungėsi prie interneto naudodami mobiliuosius įrenginius – mobiliuosius ir išmaniuosius telefonus, planšetinius kompiuterius.

Betreut.de sistemos veikimo diapazonas yra gana platus, sistema skirta naudotis keliose šalyse, o tai praplečia galimų vartotojų ratą. Visos kitos sistemos skirtos vienos šalies vartotojams.

Manopaskola.lt, raskgreitai.lt ir Taskbob sistemų veikimo logika yra panaši – vartotojas pateikia duomenis, o sistema atlieka tarpininkavimo paslaugą. Betreut.de sistema besinaudojantys vartotojai pateikia reikiamus duomenis, o sistema vartotojui – galimų paslaugų teikėjų sąrašą. Vartotojas pats turi susisiekti su paslaugų teikėju, o tai nepadeda sutaupyti laiko.

Iš lygintų sistemų tik manopaskola.lt yra visiškai nemokama. Visos kitos yra mokamos, arba dalinai mokomos. Žvelgiant iš vartotojo pusės, tai nėra pliusas – didelė dalis vartotojų dar neįprato mokėti už elektronines paslaugas.

Ne visuose lygintose sistemose informacija ar navigacija vartotojui pateikiama aiškiai ir greitai suprantamai. L. Gražytės (2006) nuomone, svetainėje ar sistemoje apsilankiusį vartotoją svarbu suinteresuoti, įtikinti, kad pasiūlymas ar sistema yra išskirtinė, ja verta naudotis. Todėl būtina atkreipti dėmesį į svetainėje ar sistemoje esančios medžiagos pateikimą.

Apibendrinant lyginamąją analizę galima teigti, kad:

1. Kuriama sistema tūri būti pritaikyta išmaniesiems įrenginiams;
2. Kurti tik mobilią aplikaciją nenaudinga, nes taip sistema prarastų tuos vartotojus, kurie neturi išmaniųjų telefonų ar mobilaus interneto;

3. Optimaliausias būdas sukurti sistemą, kuria galima būtų naudotis asmeniniuose kompiuteriuose ir išmaniuosiuose įrenginiuose – padaryti ją veikiančią per interneto naršyklę, kuri prisitaiko prie įvairių raiškų (angl. responsive). Kiekviename kompiuteryje ir išmaniajame įrenginyje yra vienokia ar kitokia interneto naršyklė, tereikia sukurti sistemos adaptaciją prie atskirų rezoliucijų. Tokios sistemos pavyzdys – Bumfix tarpininkavimo sistema. Tokių sprendimų realizacijai geriausiai tinka web programavimo kalbos;

4. Kuriamoje tarpininkavimo sistemoje informacija turi būti pateikta aiškiai, vartotojas, įsijungęs sistemą, turi iškart suprasti, kaip ja naudotis. Sukurta sistema vartotojams turi būti pasiekiama nemokamai, tai turėtų gerokai praplėsti vartotojų ratą.

2. PROGRAMAVIMO KALBOS PASIRINKIMAS KURIAMAI SISTEMAI REALIZUOTI

2.1 Programavimo kalbų populiarumas

Prieš pradedant programavimo kalbų apžvalgą, išsiaiškinsime, kokios web programavimo kalbos šiai dienai yra populiariausios, daugiausiai naudojamos, turi didžiausias programuotojų bendruomenes. Remdamiesi šia apžvalga, išskirsime keturias web programavimo kalbas, su kuriomis galima būtų atlikti kuriamos tarpininkavimo sistemos realizaciją. Žemiau esančioje lentelėje (2 lentelė) yra pateiktas programavimo kalbų populiarumas pagal www.tiobe.com. (TIOBE – 2000 metais, spalio 1 dieną, įkurta Šveicarijos įmonė, kuri specializuojasi programinės įrangos vertinime ir stebėjime, taikant plačiai naudojamus kodavimo standartus).

2.Lentelė *Programavimo kalbų populiarumas pagal www.tiobe.com.*

Vieta	**Programavimo kalba**	**Reitingas**
1.	PHP	20,528%
2.	Java	4,257%
3.	Python	2,768%
4.	Visual Basic .NET	2,561%
5.	JavaScript	2,333%
6.	Ruby	2,238%

Kitas informacijos šaltinis, kuris mums padės suformuoti šiuolaikinį požiūrį į web programavimo kalbas yra „Cleverboad‘‘ kompanijos atliktas programavimo kalbų tyrimas. (Cleverboad – tarptautinė programuotojų kompanija). Tyrimą kompanija atliko remdamasi „Google Trends‘‘ paieškos rezultatų duomenimis.

3.Lentelė *Programavimo kalbų populiarumas pagal www.cleveroad.com.*

Vieta	**Programavimo kalba**	**Reitingas**
1.	PHP	100
2.	Java	38,71

3.	Python	24,73
4.	JavaScript	21,5
5.	Visual Basic .NET	9,68
6.	Ruby	5,38

Dar vienas informacijos šaltinis, apžvelgiantis programavimo kalbų populiarumą, yra „PYPL‘‘ (PYPL logika yra tokia: kuo daugiau programavimo kalbos pamokų ieškoma, tuo programavimo kalba yra populiaresnė.)

4.Lentelė *Programavimo kalbų populiarumas pagal www.pypl.github.io*

Vieta	**Programavimo kalba**	**Reitingas**
1.	PHP	24.1 %
2.	Java	12.1 %
3.	Python	10.6 %
4.	Javascript	7.4 %
5.	Visual Basic .NET	3,1 %
6.	Ruby	2.3 %

PYPL taip pat išskyrė tris populiariausias programavimo kalbas ir pateikė jų populiarumo reitingų lentelę.

Rank	Change	Language	Share	Trend
1		PHP	24.0 %	-0.2 %
2	↑	Python	12.4 %	+1.8 %
3	↑	Java	10.6 %	-0.8 %
4		C#	8.9 %	-0.4 %
5	↑↑	Javascript	7.5 %	+0.5 %
6	↓	C++	7.4 %	-0.3 %
7	↓	C	7.2 %	+0.1 %

1 pav. **WEB programavimo kalbų populiarumo reitingų lentelė.**
Šaltinis: PYPL

Šioje lentelėje aiškiai matome, kad populiariausios, stabiliausios ir nuolat išsilaikančios populiarumo aukštumoje yra PHP, Python ir Java programavimo kalbos.

Žvelgiant į antrą, trečią ir ketvirtą lenteles, matome, kad šiai dienai populiariausia web programavimo kalba yra PHP, antrą vietą užima Java, o trečią – palyginti nesenai išpopuliarėjusi Python programavimo kalba.

2.1.1 Pasaulyje populiarių internetinių svetainių sukūrimo technologijos

Site	Up Since	Server Platform	Programming Language
Google.com	November 1998	Linux	C, Java, C++, PHP & MySQL
Facebook.com	February 2004	Linux	PHP, MySQL and C++
YouTube.com	February 2005	Linux	C, Java and MySQL
Yahoo.com	August 1995	Linux	C++, C, Java, PHP & MySQL
MSN.com (owned by Microsoft)	August 1995	Windows	ASP.net
Live.com (owned by Microsoft)	August 2008	Windows	ASP.net
Wikipedia	January 2001	Linux	PHP & MySQL
Amazon.com	October 1995	Linux & Solaris	C++, Java, J2EE
WordPress.com	November 2005	Linux	PHP & MySQL

2 pav. **Pasaulyje populiarių internetinių svetainių sukūrimo technologijos**
Šaltinis: http://www.comentum.com

Iš pasaulyje populiarių internetinių svetainių sukūrimo technologijų paveikslėlio matyti, kad daugiausiai nagrinėjamų projektų yra sukurti naudojant PHP programavimo kalbą ir ASP.net. dinaminių tinklalapių struktūros kūrimo technologiją.

Iš 2.1 ir 2.1.1 poskyrių galime daryti išvadą, kad populiariausios ir plačiausiai šiai dienai naudojamos web programavimo kalbos yra šios: PHP, Java, Python, ir ASP.net. (dinaminių tinklalapių kūrimo technologija). Kitame skyriuje trumpai apžvelgsime šias programavimo kalbas ir išsirinksime vieną, tinkamiausią, web programavimo kalbą kuriamai tarpininkavimo sistemai realizuoti.

2.2 Programavimo kalbų apžvalga

Java – objektiškai orientuota programavimo kalba, 1991 m. sukurta Džeimso Goslingo ir kitų Sun Microsystems kompanijos inžinierių. Apie ją oficialiai paskelbta 1995 m. gegužės 23 d., o išleista ji tų pačių metų lapkritį. Pradžioje, kuriant Java kalbą, buvo planuota naudoti ją buities įrenginių mikrokontrolerių programavimui. Todėl Java kalba (tuo metu vadinta Oak) nuo pat pradžių buvo kuriama taip, kad būtų nepriklausoma nuo architektūros, kompaktiška ir saugi. Tačiau tuo metu Java nebuvo tokia populiari ir net pasirodė esanti ne itin reikalinga. Didžiausio pripažinimo ji sulaukė tada, kai paplito pasaulinio tinklo (angl. *World Wide Web*) naudojimas ir jau visai kitoje – interneto srityje (J. Katina, 2009).

PHP (angliškų žodžių akronimas : *Hypertext Preprocessor*) – plačiai paplitusi dinaminė interpretuojama programavimo kalba. Pačioje pradžioje ji buvo orientuota tik į internetą, tačiau dabar ja galima programuoti ne tik internetines svetaines. Pavyzdžiui, su „php–gtk“ galima kurti pilnavertes „cross“ platformines programas ir t.t. PHP skriptai yra interpretuojami ir įvykdomi serverio pusėje (Sinkevičius, 2008).

Python – interpretuojama interaktyvi programavimo kalba, sukurta Guido van Rossumo, 1990 m. Pirmiausiai tai buvo scenarijų kalba AmoebaOS operacinei sistemai. Ši programavimo kalba dažniausiai lyginama su Perl, Java ir Ruby. Python yra atviro kodo, daugialypė programavimo kalba – ji leidžia naudoti keletą programavimo stilių: objektinį, struktūrinį, funkcinį, aspektinį. Python naudoja dinaminį tipų tikrinimą. Šios kalbos kūrėjų tikslas buvo sukurti kalbą, kuri būtų lengvai skaitoma, išraiškinga, išreikštinė, paprasta. Nors pradžioje ji buvo kuriama kaip scenarijų kalba, dabar ši programavimo kalba naudojama ir dideliems programiniams projektams realizuoti. Tai aukšto lygio dinaminė programavimo kalba. Python yra interpretuojama, panašiai kaip Java (programa pirma kompiliuojama į baitinį kodą, o paskui virtuali mašina tą kodą interpretuoja). Skirtingai nuo Java, Python kompiliavimas atliekamas automatiškai, paleidus interpretatorių (E. Matijošius, 2008).

ASP.NET – yra Microsoft serverio pusės programavimo technologija, skirta kurti dinamiškai generuojamus „web“ puslapius, kurie pažymėti kaip Interneto Informacijos Serviso (IIS) papildai. ASP suteikia galimybę sujungti HTML puslapius, skriptų

komandas bei sukurti interaktyvius „web“ puslapius arba „web“ pagrindu paremtas programas, kurios lengvai modifikuojamos (Mahanata, 2005).

2.3 Programavimo kalbų palyginimas

Šiame skyriuje palygintos PHP, Java ir Python programavimo kalbos, taip pat dėl savo populiarumo ir plataus naudojimo į palyginimą įtrauka ASP.NET dinaminių tinklalapių struktūros kūrimo technologija.

2.3.1 Programavimo kalbų palyginimo kriterijų pasirinkimas

Norint apibūdinti programavimo kalbas, reikia iš anksto susitarti dėl jų vertinimo kriterijų. Suprantama, kad gali būti labai daug savybių, kurių tikimės iš programavimo kalbų. Reikia atsirinkti svarbiausias, o renkantis konkrečią programavimo kalbą, pagalvoti apie sritį, kurioje ji bus taikoma.

Pateiksime keturias pagrindines programavimo kalbų savybes, kurias nurodo R.W. Sebesta (2008), laikytinas kalbų vertinimo kriterijais:

1. programų skaitomumas,
2. programų rašymo patogumas,
3. patikimumas.
4. išlaidos (programoms kurti, programuotojams mokytis ir t.t.)

Remiantis šiomis pateiktomis programavimo kalbų savybėmis, taip pat atsižvelgiant į kuriamo projekto sritį, programavimo kalboms palyginti pasirinkti šie vertinimo kriterijai:

Paskutinė versija, atsiradimo metai – kada programavimo kalba atsirado, kiek versijų yra išleista. Šis vertinimo kriterijus pasirinktas atsižvelgiant į programavimo kalbų patikimumo vertinimą.

Paskirtis – kam programavimo kalba skirta (internetiniams projektams realizuoti, konsolinėms programoms kurti, kaip daugiafunkcinė programavimo kalba ir t.t.).

Licencijavimas – mokama, nemokama, ar dalinai mokama.

Vartotojų paskyrų skaičius *(GitHub paskyrų duomenimis)* – programavimo kalbos bendruomenės narių skaičius. Kuo programavimo bendruomenės ratas platesnis, tuo programavimo kalba populiaresnė, lengviau perprantama, greičiau sprendžiamos iškilusios problemos programavimo procese.

Kitų programavimo kalbų įdiegimas – kokias kitas programavimo kalbas galima integruoti.

Programavimo aplinkos – ar programavimo kalba turi programavimo aplinkų pasirinkimą, kokios aplinkų prieinamumo galimybės (mokama / nemokama).

Ar turi sukurtų programavimo karkasų – programavimo kalbos karkasai supaprastina ir palengvina projektų kūrimą, todėl renkantis programavimo kalbą reikia atsižvelgti, ar ji turi sukurtų programavimo karkasų.

Objektiškai orientuota – ar programavimo kalba palaiko objektinį programavimą. Šis programavimo kalbos kriterijus yra svarbus projektuojant ir programuojant projektą.

Standartizacija – ar programavimo kalba yra standartizuota, ar ne. Šis vertinimo kriterijus pasirinktas, atsižvelgiant į programavimo kalbų patikimumo vertinimą.

5.Lentelė *Web Programavimo kalbų palyginimas.*

	PHP	**Java**	**Python**	**ASP.NET**
Paskutinė versija	7.0.0	8	3.5.1	5
Atsiradimo metai	1995	1995	1991	2002
Paskirtis	Tik Web programavimas	Android platformos programų kūrimas (apps), web programavimas	Universali programavimo kalba	Tik Web programavimas
Licenzijavimas	Nemokama	Nemokama	Nemokama	Mokama
Vartotojų paskyrų skaičius *(GitHut paskyrų duomenimis)*	1,391,467	2,323,315	1,654,226	124,818
Programavimo aplinkos	Dauguma nemokamų	Dauguma nemokamų	Dauguma nemokamų	Pagrindinė IDE Visual Studio programa, kuri yra mokama.
Kitų programavimo kalbų įdiegimas	C, PERL, JAVA, C++, TCL	C, C++	C, C++, Java	C#, Visual Basic.Net, Jscripy, J#
Statinė / dinaminė	Dinaminė	Statinė	Dinaminė	Dinaminė
Ar turi sukurtų programavimo karkasų	Taip	Taip	Taip	Ne
Objektiškai orientuota	Taip	Taip	Taip	Taip
Standartizacija	Nestandartizuota	Standartizuota	Nestandartizuota	Nestandartizuota

Iš lygintų programavimo kalbų pritaikytos tik WEB programavimui yra PHP ir ASP.net programavimo kalbos. ASP.net kalba yra mokama, PHP – atviro kodo. Šiuo aspektu, pranašesnė yra PHP programavimo kalba. Objektiškai orientuotos yra visos programavimo kalbos, tačiau iš jų paruoštus karkasus (angl. framework) turi PHP, Java ir Python kalbos.

Vartotojų paskyrų skaičiumi (GitHub paskyrų duomenimis) pirmauja PHP ir Java programavimo kalbos. Vienintelė nagrinėta programavimo kalba Java yra statinė, o ne dinaminė. Kuriamai internetinei tarpininkavimo sistemai tinkamesnė yra dinaminė programavimo kalba.

Iš 2.1, 2.1.1, 2.3 poskyrių galima daryti išvadą, kad projektui kurti tinkamiausia programavimo kalba, turinti daugiausiai pranašumų įvairiais aspektais, yra PHP programavimo kalba.

Siekiant projekto kūrimo darbus optimizuoti, bus naudojamas karkasas (angl. framework). Kitame poskyryje apžvelgsime išsirinktos programavimo kalbos (PHP) karkasus, juos palyginsime ir išrinksime tinkamiausią projektui kurti.

2.4 PHP programavimo kalbos karkasų apžvalga

Karkasai yra pagrindas, reikalingas tvarkingam programuotojų rašomų kodų išdėtymui. Dokumentacija, dažnai pasitaikančios problemos, sprendimai, patarimai ir konsultacijos pateikiamos internete atviro kodo stiliumi. Prieš karkasų naudojimą, didžioji dalis internetinių projektų buvo kuriama personalizuotai, kiekvienas programuotojas naudojo savo asmeninį technikos ir metodų arsenalą, tvarkant kodo išdėstymą ir kuriant sistemos architektūrą. Tai kėlė daug problemų: dokumentacijos trūkumas, sudėtingi karkasai ir bendras klausimas, kas bus kai pasikeis programuotojas. (M. Vaitkūnas, 2013)

Karkasai buvo pradėti naudoti 2004 metais, kai buvo pristatyti kaip visapusiškas sprendimas, kuriant stabilius internetinius projektus. Populiariausi PHP karkasai daugiausia buvo skirti projektavimo metodologijai, buvo naudojamas modelio–vaizdo–kontrolierius (MVC) (R., Woler, 2012). Plačiau MVC šablonus apžvelgsime 2.4.1 poskyryje.

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

Paprastai renkantis karkasą yra atsižvelgiama į šiuos kriterijus (M.Vaitkūnas, 2013) :

- ✓ Populiarumas;
- ✓ Dokumentacijos aiškumas;
- ✓ Greitis;
- ✓ Vartotojų atsiliepimai internetinėje erdvėje;
- ✓ Karkaso paprastumas;
- ✓ Lankstumas.

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

Šiuo metu vieni iš dažniausiai naudojamų karkasų yra šie:

- Laravel – žaibiškai išpopuliarėjo prieš maždaug 2–3 metus dėl savo puikios struktūros, įrankių ir semantiškų funkcijų. Šiuo metu tai yra vienas iš populiariausių (galbūt net pats populiariausias) karkasų mažiems – vidutiniams projektams. Jame ypatingas dėmesys telkiamas naudojimo patogumui ir paprastumui.
- Symfony2 – Symfony2 karkasai dažniausiai pasirenkami kuriant didesnius, komercinius projektus. Pats Symfony2 karkasas pakankamai populiarus, tačiau daugiausiai populiarumo yra sulaukę jo sukurti komponentai.
- Zend Framework 2 – panašiai kaip ir Symfony2, Zend karkasai labiau naudojami dideliems, komerciniams projektams. Tokie karkasai reikalauja daugiau serverio išteklių. Zend karkasas ateina su daugybe Zend bibliotekų, kurias galima panaudoti universaliai. Apskritai kompanija Zend yra plačiai susijusi su PHP vystymu, pvz. pati PHP aplikacija naudoja Zend variklius ir bibliotekas. Zend Framework yra labai galingas, tačiau ir pakankamai sudėtingas, daug žinių reikalaujantis karkasas.

2.4.1 MVC šablonas

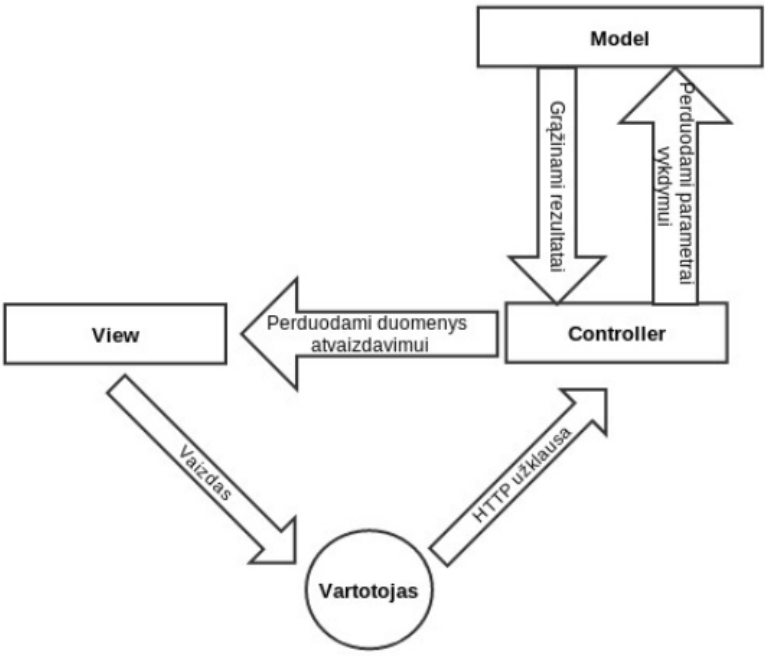
MVC – tai Model–View–Controller šablonas arba principas, vartotojui pateikiamą informaciją atskiriantis nuo sąveikos su ja. Šis šablonas padalina programą į trijų rūšių komponentus ir apibrėžia sąveiką tarp jų.

MVC šabloną sudaro trys elementai:

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

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

Vizualiai atvaizduotą MVC schemą galima pamatyti žemiau pateiktame paveikslėlyje (Pav. 3).



3 pav. **MVC šablono grafinis vaizdas**
Šaltinis: M. Vaitkūnas. PHP karkasų architektūrinis tyrimas.

2.5 PHP programavimo kalbos karkasų palyginimas

PHP karkasų vertinimo kriterijai pasirinkti, atsižvelgiant į karkasų kūrėjų pateiktas dokumentacijas. Renkantis palyginimo aspektus buvo atsižvelgta ir į „Social Compare‘‘ palyginimo sistemos pateiktus karkasų vertinimo kriterijus.

Trumpai aptarsime pasirinktus aspektus:

Paskutinė versija – nurodoma, kiek versijų karkasas turi ir kuri versija yra naujausia.

Licencija – nusako, kokias naudojimo teises suteikia karkaso kūrėjas.

PHP reikalavimai – pateikiami minimalūs programavimo kalbos reikalavimai (nurodoma minimali versija). Tai pakankamai svarbus kriterijus konfigūruojant serverį.

Kodo generavimas – ar karkasas palaiko funkciją, kad dalį kodo programuotojas galėtų susigeneruoti, pavyzdžiui, per komandinę eilutę pateikęs tam tikras komandas. Ši funkcija pagreitina programuotojo darbą.

ORM – Sql užklausų pateikimo įrankis.

Templeitų sistema – karkaso įrankis, leidžiantis struktūrizuotai programuoti kuriamos sistemos išorinės dalies atvaizdavimą.

Testavimų biblioteka – karkasas turi integruotus testavimo įrankius.

Meniu generavimas – karkasas turi integruotus vartotojo meniu kūrimo įrankius.

CRUD – ar karkasas turi CRUD (kurti, skaityti, atnaujinti, ištrinti duomenų bazės įrašus) operacijų generavimo įrankį.

Autorizacijos modulis – ar karkase yra integruotas autorizacijos modulis. Šis modulis programuotojams palengvina vartotojų prisijungimo/atsijungimo, teisių priskyrimo funkcijų valdymą.

Valdymo panelė – ar karkasas turi integruotą valdymo panelę, kuria galima butų keisti sistemos informaciją, karkaso konfigūraciją ir t.t.

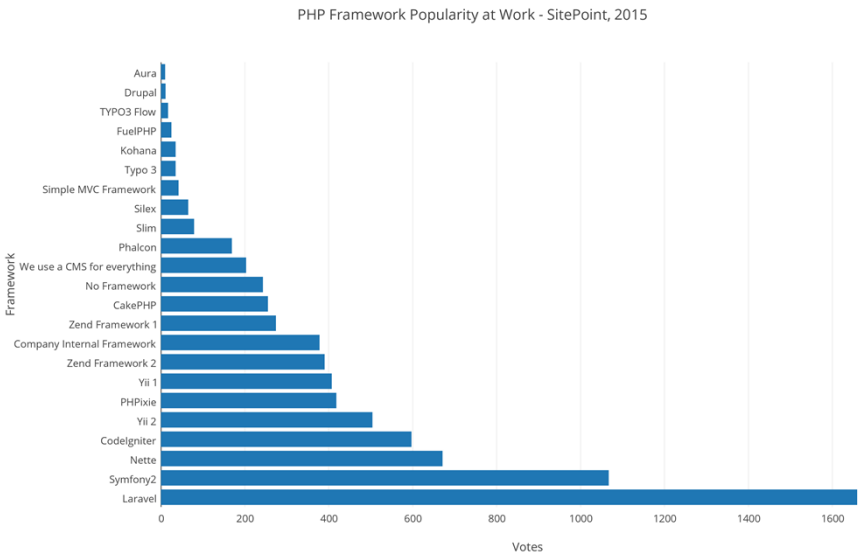
Galimybė dirbti su keliomis duomenų bazėmis – ar karkasas turi galimybę vienu metu jungtis prie keleto duomenų bazių. Ši funkcija dažniausiai reikalinga kuriant projektus su didelėmis duomenų bazėmis.

Naudojimo paprastumas – kaip lengvai galima programuoti karkasą, kokio lygio pateikta dokumentacija, kiek plati programuotojų bendruomenė. Tai pakankamai svarbus rodiklis, nulemiantis kuriamo projekto programavimo spartą ir kokybę.

6.Lentelė *PHP karkasų palyginimas.*

	Laravel	**Symfony2**	**Zend Framework 2**
Paskutinė versija	5.1	3.0	2.4.9
Licencija	Atviro kodo (MIT)	Atviro kodo (MIT)	Atviro kodo (BSD)
PHP reikalavimai	PHP >= 5.5.9	PHP >= 5.3.9	PHP >= 5.4
Kodo generavimas	CLI	CLI	–
ORM	ELOQUENT ORM	Doctrine 2, Propel	–
Templeitų sistema	Blade, PHP, Custom	PHP, Twig	–
Testavimų biblioteka	PHPUnit	PHPUnit	PHPUnit
Meniu generavimas	–	–	–
CRUD generavimas	–	SensioGeneratorBundle	–
Autorizacijos modulis	+	+	–
Valdymo panelė	–	–	–
Galimybė dirbti su keliomis duomenų bazėmis	+	+	–
Naudojimo paprastumas *(vertinama, remiantis programuotojų bendruomenės nuomonėmis)*	Įvertinimas: 5/5	Įvertinimas: 3/5	Įvertinimas: 3.5/5

PHP karkasų palyginimas pagal populiarumą:



4 pav. **PHP karkasų populiarumas**
Šaltinis: http://www.sitepoint.com

Iš PHP karkasų palyginimo lentelės matome, kad daugiausia privalumų turintys karkasai yra Laravel ir Symfony2. Zend Framework 2 karkasas savo funkcionalumu šiek tiek atsilieka. Kaip minima 2.3 poskyryje, šis karkasas skirtas kurti didelės apimties projektams, didžiąją dalį reikiamų funkcijų ar įrankių, naudodamas karkasą, turi susikurti pats programos kūrėjas.

Laravel karkasas lenkia Symfony2 savo paprastumu, jį galima nesudėtingai perprasti ir greitai pradėti kurti realų projektą. Symfony2 karkasas yra sudėtingesnis, o jų galimybės panašios.

Laravel karkasas gerokai lenkia kitus PHP karkasus savo populiarumu – šiai dienai jis yra pats populiariausias, labiausiai programuotojų bendruomenės pripažintas karkasas ir jo populiarumas vis auga. Dirbant su šiuo karkasu galima tikėtis didelio palaikymo, pagalbos iš programuotojų bendruomenės, sprendžiant iškilusias problemas, taip pat šis karkasas turi gerai parengtą dokumentaciją. Iš šio poskyrio galima daryti išvadą, kad labiausiai tinkamas PHP karkasas projektui kurti yra Laravel.

3. DUOMENŲ BAZIŲ ANALIZĖ

3.1 Duomenų bazių apžvalga

Kuriamo projekto duomenys bus dinaminiai (nuolat keisis, bus papildomi, kursis nauji įrašai), dažniausiai tokie duomenys saugomi duomenų bazėje.

Šiame poskyryje palyginsime kelias populiariausias duomenų bazių platformas. Jų populiarumas nustatytas remiantis žemiau pateikto šaltinio duomenimis.

Rank			DBMS	Database Model	Score		
Apr 2017	Mar 2017	Apr 2016			Apr 2017	Mar 2017	Apr 2016
1.	1.	1.	Oracle	Relational DBMS	1402.00	+2.50	-65.54
2.	2.	2.	MySQL	Relational DBMS	1364.62	-11.46	-5.49
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1204.77	-2.72	+69.72

5 pav. **Duomenų bazių populiarumas**
Šaltinis: DB–Engines Ranking, 2017

Oracle – viena didžiausių ir dažniausiai naudojamų reliacinių duomenų bazių valdymo sistemų. Oracle yra labai plačiai naudojama bankinėse, finansinėse ir mokslinėse sistemose duomenims saugoti, apdoroti ir analizuoti.

Microsoft SQL Server – yra duomenų bazių valdymui ir analizei skirta sistema. Naudojama e–komercijai, gamybos linijoms ir duomenų sandėliavimo sprendimams. SQL Server taip pat tinka vidaus infrastruktūrai, „Debesų“ ar nevienalytėms aplinkoms.

MySQL – viena iš reliacinių duomenų bazių valdymo sistemų, palaikanti daugelį naudotojų, dirbanti SQL kalbos pagrindu. MySQL yra atviro kodo programinė įranga (GPL ir kitos licencijos), vystoma ir palaikoma švedų kompanijos „MySQL AB“.

Remiantis (E. Ivanauskienė, 2006) moksliniu darbu pasirinkti šie duomenų bazių palyginimo kriterijai:

Tipas – duomenų bazių tipas nurodo, kaip duomenų bazėje yra saugoma informacija.

Kūrėjas – nurodo, kas yra duomenų bazės autorius. Šis kriterijus taip pat nusako duomenų bazės patikimumą.

Paskutinė versija – nurodo, kada paskutini kartą buvo išleista naujausia versija. Remiantis šiuo kriterijumi, galime spręsti apie duomenų bazės palaikymą.

Licencija – nusako, ar duomenų bazės platforma mokama, ar ne.

Sukūrimo kalba – kokia programavimo kalba sukurta duomenų bazės platforma. Tai pakankamai svarbus kriterijus, jei kuriamas projektas bus plečiamas ir bus reikalingos korekcijos duomenų bazės platformos variklyje.

Serverio platforma – kokią serverio operacinę sistemą palaiko duomenų bazė.

XML schemos – ar duomenų bazės platforma turi galimybę importuoti ar eksportuoti duomenys XML formatu. Tai pakankamai svarbus kriterijus, migruojant tarp duomenų bazių.

Suprantamos programavimo kalbos – kokiomis programavimo kalbomis galima jungtis prie duomenų bazės, pateikti užklausas, gauti duomenis.

Transakcijų palaikymas – ar duomenų bazės platforma turi galimybę aprašyti numatytus veiksmus – transakcijas. Transakcijos pavyzdys gali būti veiksmai, kuriuos reikia atlikti, norint kliento pinigus iš vienos sąskaitos pervesti į kitą.

3.2 Duomenų bazių palyginimas

7.Lentelė *Duomenų bazių platformų palyginimas*.

	Microsoft SQL Server	**MySQL**	**Oracle**
Tipas	Reliacinė	Reliacinė	Reliacinė
Kūrėjas	Microsoft	Oracle	Oracle
Paskutinė versija	SQL Server 2015	5.7.9	12.1.0.2
Licencija	Komercinė	Atviro kodo	Komercinė
Sukūrimo kalba	C++	C ir C++	C ir C++
Serverio platforma	Windows	FreeBSD Linux OS X Solaris Windows	AIX HP–UX Linux OS X Solaris Windows z/OS
XML schemos	+	+	+
Suprantamos programavimo kalbos	.Net Java PHP Python Ruby Visual Basic	Ada C C# C++ D Eiffel Erlang Haskell Java Objective–C OCaml Perl PHP Python	C C# C++ Clojure Cobol Eiffel Erlang Fortran Groovy Haskell Java JavaScript Lisp Objective C

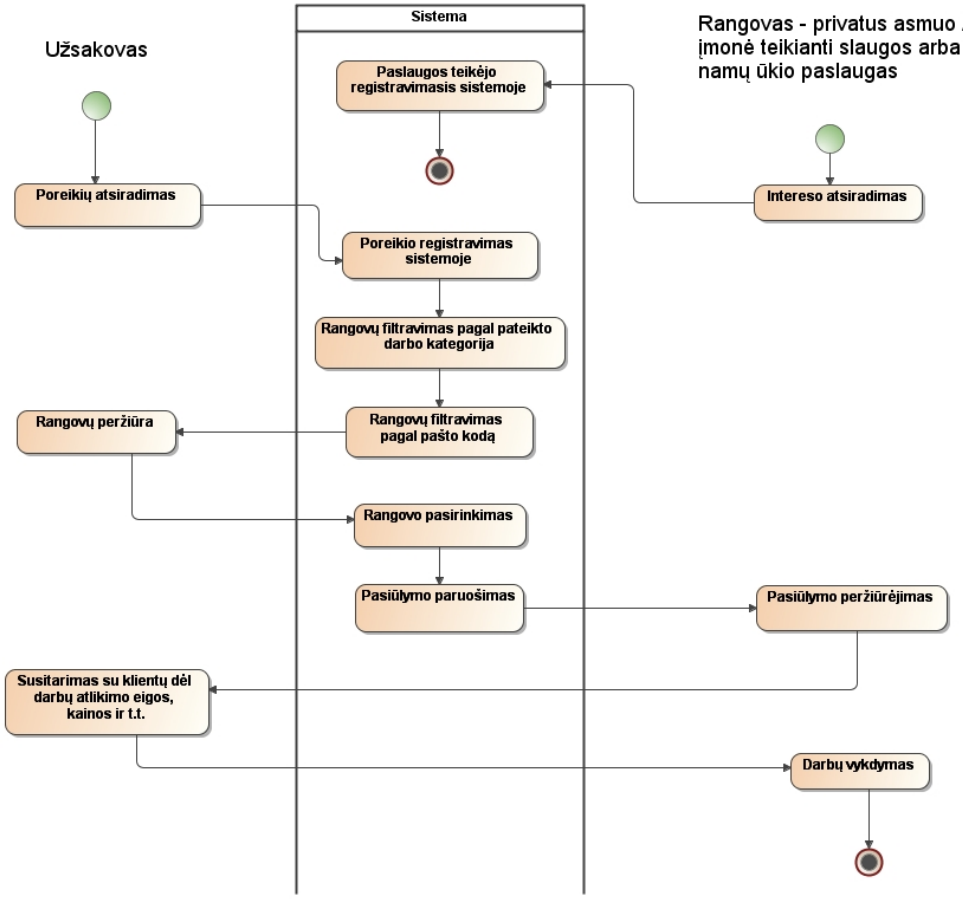
		Ruby Scheme Tcl	OCaml Perl PHP Python R Ruby Scala Tcl Visual Basic
Transakcijų palaikymas	+	+	+

Iš trijų lygintų duomenų bazių platformų nemokama yra tik MySQL. Oracle ir Microsoft SQL Server duomenų bazės yra komercinės, Microsoft SQL Server veikia tik Windows operacinėje sistemoje, kuri taip pat yra mokama. Kuriamas projektas bus PHP programavimo kalba, kurią palaiko visos lygintos duomenų bazės. XML schemas ir transakcijas palaiko visos lygintos duomenų bazės.

Kadangi kuriamas projektas duomenų bazėje nesaugos itin svarbių, slaptų duomenų, jam galima rintis nemokamą, visus reikalavimus atitinkančią duomenų bazę – MySQL.

4. TARPININKAVIMO SISTEMOS MODELIO PROJEKTAS

4.1 Esamų tarpininkavimo sistemų principinės veikimo schemos ir jų apžvalga



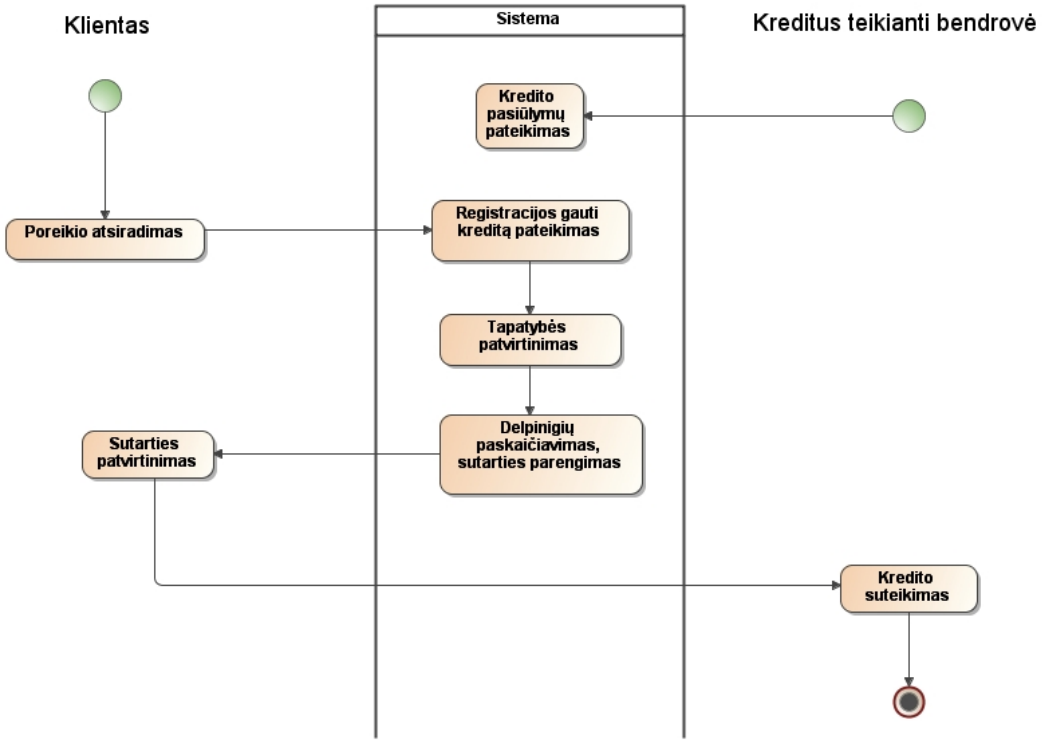
6 pav. **Virtualaus tarpininko Betreut.de paslaugų teikimo principinė schemą**
Šaltinis: Sudaryta autoriaus

Betreut.de paslaugų teikimo principinėje schemoje matyti, kad tarpininkavimo sistemos vaidmuo yra klientų duomenų bazės kaupimas, užsakovų darbų registracija, pasiūlymų teikimas rangovams. Schema iliustruoja, kaip sistema, prieš siunčiant pasiūlymus rangovams, juos filtruoja pagal pateiktos užduoties kategorijas, kitas žingsnis – filtravimas pagal pašto kodą.

Šiame rangovų atrankos modelyje galima įžvelgti dviejų lygių atrankos mechanizmą: pagal darbų kategorijas ir pagal rangovo pašto kodą. Galime daryti

prielaidą, kad atranka pagal pašto kodą reikalinga norint rasti arčiausiai esančius meistrus.

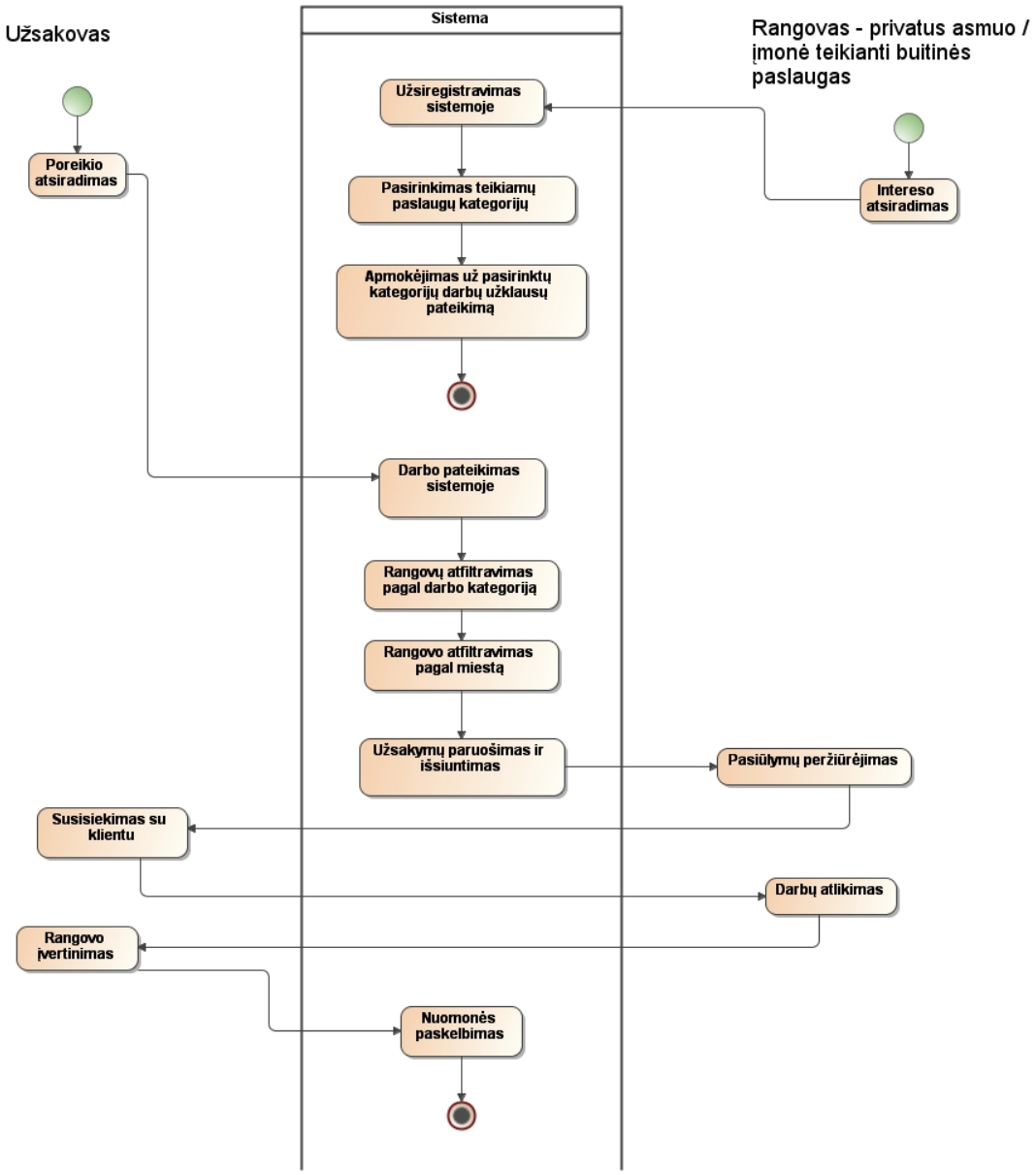
Tolimesnėje kliento ir rangovo komunikacijoje sistema nedalyvauja.



7 pav. **Virtualaus tarpininko manopaskola.lt paslaugų teikimo principinė schemą**
Šaltinis: Sudaryta autoriaus

Pagrindinė manopaskola.lt sistemos paskirtis yra paskolą pasiimti norintį klientą nukreipti į kreditus teikiančią bendrovę. Šį sistema skiriasi nuo kitų nagrinėtų elektroninių tarpininkų, nes joje rangovas yra tik vienas – bendrovė, teikianti paskolas.

Pateiktos sistemos veikimo principinėje schemoje matyti, kad tarpininkavimo mechanizmo pagrindinės užduotys yra užregistruoti klientą, suskaičiuoti būsimus delspinigius, parengti paskolos teikimo sutartį. Šis tarpininkavimo modelis skiriasi nuo kitų nagrinėtų tuo, kad jame nėra atrankos mechanizmo.



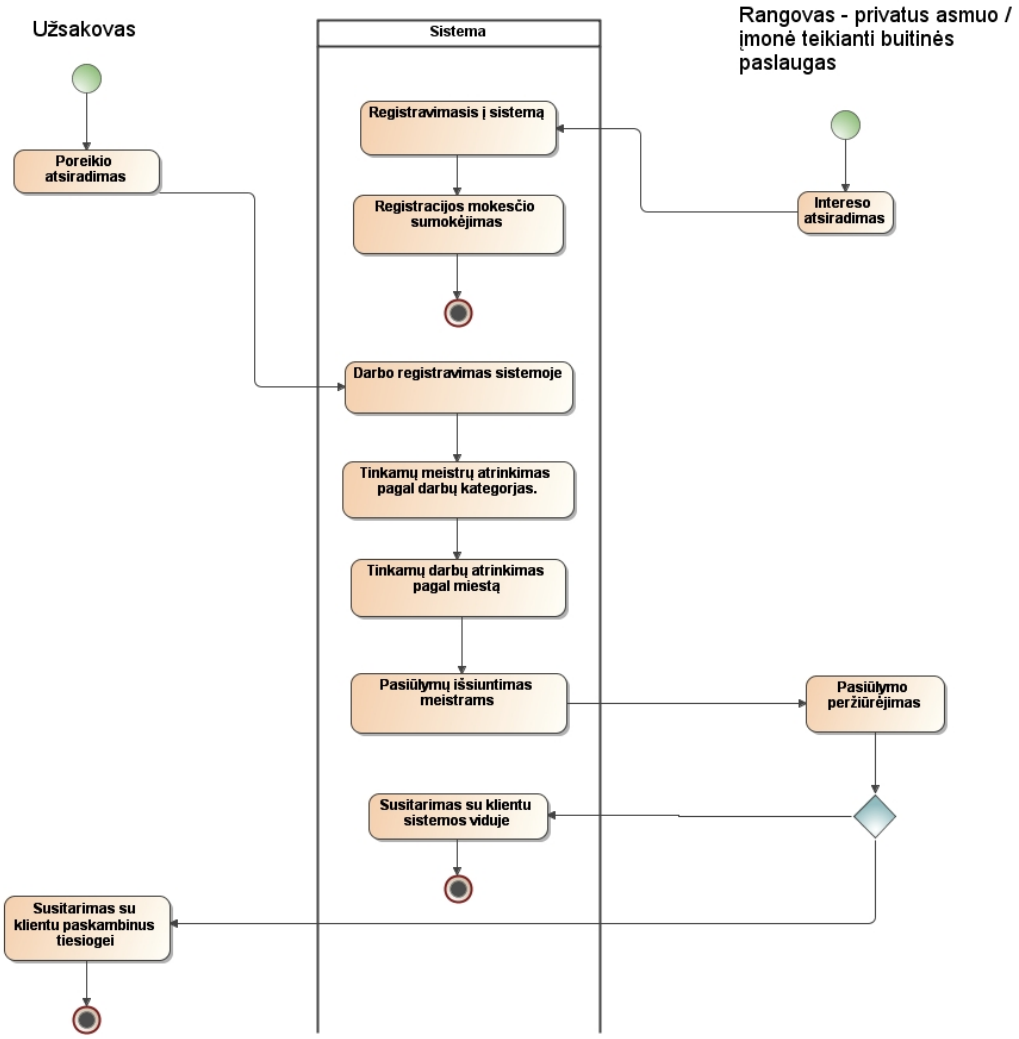
8 pav. **Virtualaus tarpininko raskgreitai.lt paslaugų teikimo principinė schema**
Šaltinis: Sudaryta autoriaus

Raskgreitai.lt tarpininkavimo sistemos principinėje schemoje matyti, kad rangovas turi susimokėti registracijos mokestį, kitaip sistema jam neleis baigti registracijos.

Šis tarpininkavimo modelis turi du rangovų atrankos lygius: pagal darbų kategoriją ir pagal miestą.

Atrinkus tinkamus rangovus, sistema jiems išsiuntinėja pasiūlymus, tada sistema savo darbą baigia, o rangovai su užsakovu užsakymo detales aiškinasi patys (telefonu).

Užsakovas išsirenka tinkamą meistrą. Užsakovas sistemoje turi galimybę parašyti atsiliepimą apie atliktą darbą, skirti reitingo taškus.



9 pav. **Virtualaus tarpininko Taskbob paslaugų teikimo principinė schema**
Šaltinis: Sudaryta autoriaus

Šioje sistemoje, kaip ir raskgreitai.lt, virtualioje tarpininkavimo sistemoje rangovas registracijos metu turi sumokėti registracijos mokestį.

Ši tarpininkavimo sistema, kaip ir raskgreitai.lt, turi du rangovų atrankos lygius: atranka pagal miestą ir atranka pagal meistrų kategorijas. Kaip matome „Taskbob‘‘ sistemos principinėje schemoje, sistema turi vidinę meistro ir užsakovo bendravimo galimybę. Meistrai, gavę darbų pasiūlymus, gali su užsakovais tartis per tarpininkavimo

sistemą arba skambinti tiesiogiai. Sistema po meistro ir užsakovo suvedimo baigia savo darbą.

Apžvelgus nagrinėtų virtualių tarpininkavimo sistemų principines schemas, galima daryti šias išvadas:

1. nagrinėtos tarpininkavimo sistemos sudarytos iš šių esminių komponentų: rangovų registracija, užsakymų pateikimas, rangovų atrankos mechanizmas, pasiūlymų pateikimas rangovams;
2. nagrinėtose tarpininkavimo sistemose svarbiausią vaidmenį atlieka rangovų atrankos mechanizmas;
3. nagrinėtų tarpininkavimo sistemų rangovų atrankos mechanizmai yra sudaryti iš dviejų dalių: atranka pagal miestą ir atranka pagal darbų kategoriją.

Analizuotų sistemų pagrindinis veikimo trūkumas tas, kad sistemos atrenka meistrus tik pagal pasirinktą miestą ir darbų kategoriją, o tai pakankamai primityvus atrankos modelis.

Didžioji dalis buitinių paslaugų yra smulkūs darbai, kuriuos atlikti meistrams nereikia daug laiko, jų įkainiai taip pat nėra dideli. Būtų racionalu, jei tokiems darbams atlikti sistema parinktų arčiausiai užsakovo esančius meistrus. Šitaip klientui nereikėtų ilgai laukti meistro, o meistras išvengtų situacijų, kai uždarbis už atliktą darbą beveik neviršija kelionės iki kliento kaštų.

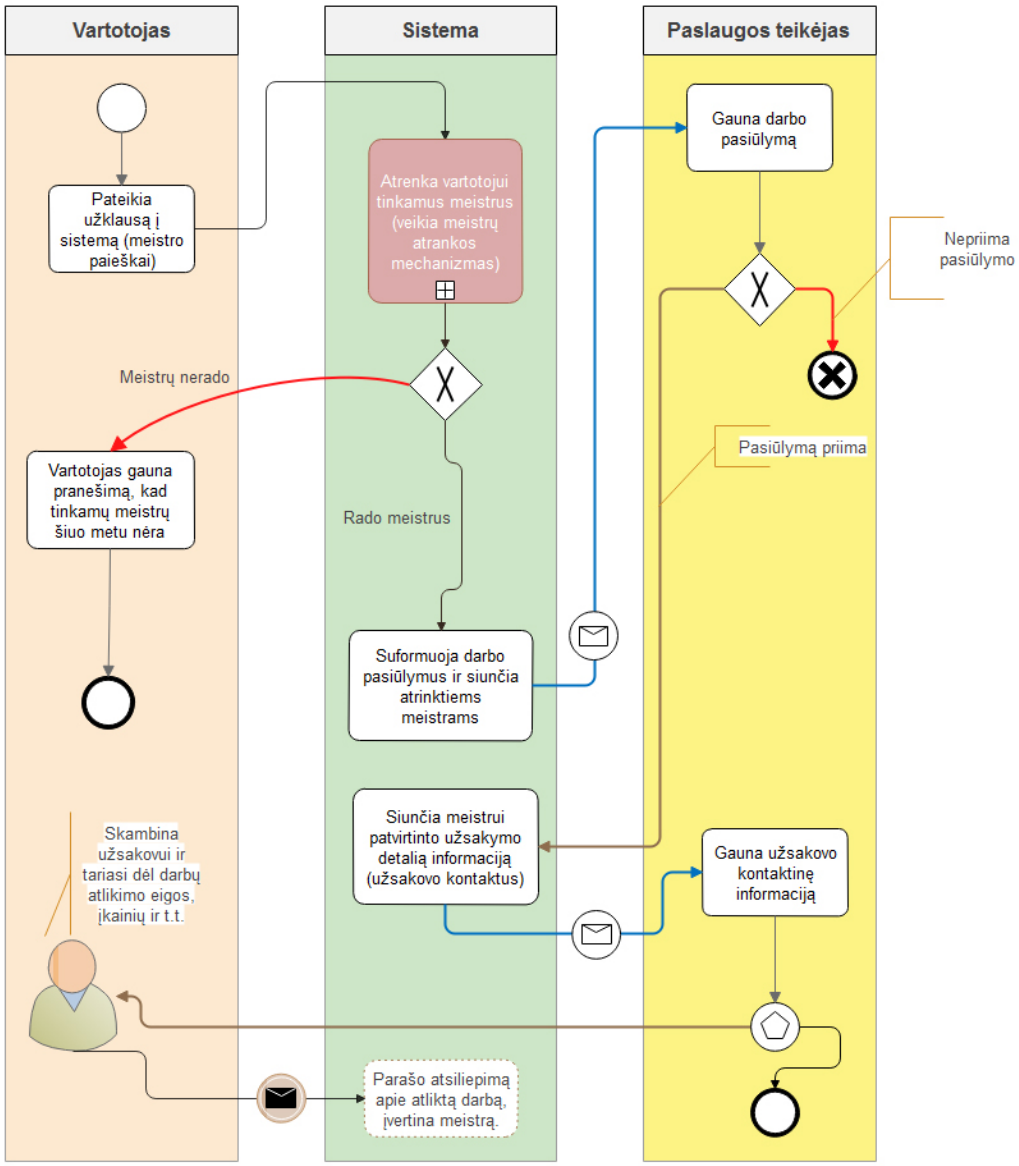
4.2 Tarpininkavimo sistemos BPMN modelis

Išanalizavus virtualių tarpininkavimo sistemų principines schemas, išsiaiškinus pagrindinius sistemų komponentus, jų trūkumus suprojektuotas kuriamos sistemos modelis.

Šiame modelyje pavaizduotas abstraktus sistemos veikimas, nuo kliento užklausos pateikimo, iki darbo atlikimo ir meistro įvertinimo. Modelyje nėra aptartos tokios detalės, kaip duomenų pateikimas, registracija ir t.t. Šio modelio paskirtis – pavaizduoti sistemos veikimo principą.

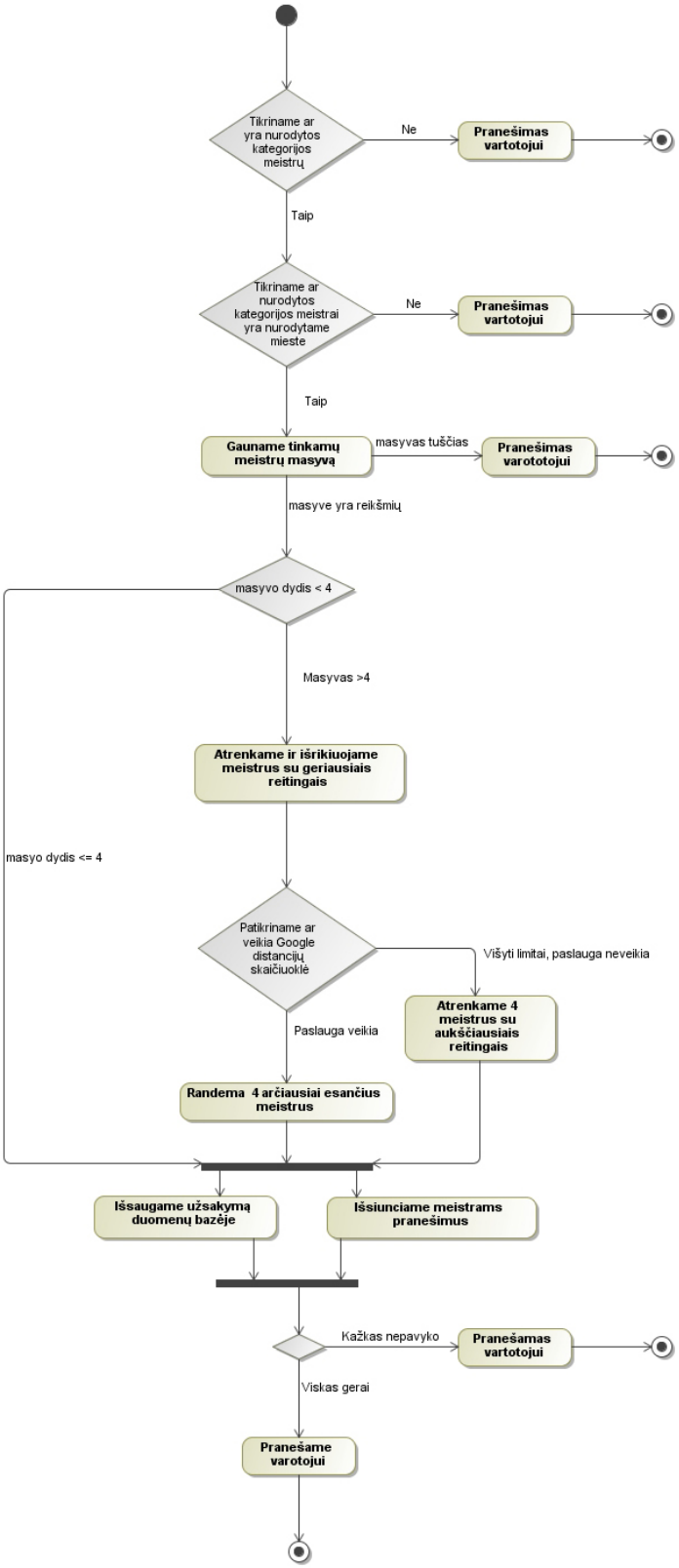
Modelis suskirstytas į tris dalis: vartotojas, sistema, paslaugos teikėjas – tai pagrindiniai sistemos dalyviai. Tarp jų esančios rodyklės iliustruoja informacijos apsikeitimą, įvykių priskyrimą. Modelyje rodyklės pažymėtos skirtingomis spalvomis:

- raudona spalva – procesas nutrūksta
- mėlyna spalva – sistemos informacijos pateikimas paslaugos teikėjui
- juoda spalva – tarpiniai veiksmai
- ruda spalva – paslaugos teikėjo atsakas



10 pav. **Būsimos tarpininkavimo sistemos grafinis modelis**
Šaltinis: Sudaryta autoriaus

4.3 Tarpininkavimo sistemos meistrų atrankos algoritmas



11 pav. **Būsimos tarpininkavimo sistemos meistrų atrankos algoritmas**
Šaltinis: Sudaryta autoriaus

Kuriamos sistemos meistrų paieškos mechanizmo pagrindinis pranašumas ir unikalumas yra meistrų atranka pagal darbų kategorijas, miestą ir atstumą iki kliento. Atranka pagal atstumą vykdoma atsižvelgiant į meistrų vertinimo rodiklį. Sistema pirmiausia išrenka reikiamus patikimiausius meistrus, o tada iš jų – esančius arčiausiai.

Sistema naudojasi kompanijos Google teikiama paslauga – atstumo nustatymu tarp dviejų taškų.

Šios atrankos mechanizmo pagrindinis trūkumas yra tai, kad naudojami nutolusios sistemos teikiami duomenys, todėl pasitaiko, kad atstumų nustatymo paslauga būna nepasiekiama. Tokias atvejais, kad sistema nepasimestų, suprojektuotas algoritmas pirmiausia patikrina, ar veikia atstumų nustatymų paslauga. Jei paslauga nepasiekiama, sistema išsiunčia darbų pasiūlymus meistrams, kurie atitinka nurodytą darbų kategoriją, miestą ir turi aukščiausią patikimumo reitingą. Patikimumo reitingus sudaro užsakovai, kurie meistrą įvertina po paslaugos atlikimo.

Algoritmo pranašumai, lyginant su analizuotų sistemų paieškos mechanizmais: tikslesnė meistrų atranka, randami arčiausiai esantys patikimiausi meistrai.

Algoritmo pranašumai, lyginant su kitais paieškos algoritmais: stabilumas, algoritmas atlieka savo darbą, net jei atstumų nustatymo paslauga neveikia, tokiu atveju algoritmas pakeičia savo meistrų atrankos kriterijus.

4.4 Tarpininkavimo sistemos prototipo testavimas

Testavimą galima apibūdinti kaip procedūrą, per kurią siekiama nustatyti sistemos atitikimą veiklos reikalavimams (I. Plauska, 2013). Bendru atveju testavimas reikalingas dėl keturių priežasčių (A. Berger, 2013):

- Siekiant rasti klaidas sistemoje (testavimas – vienintelis būdas tai pasiekti);
- Siekiant sumažinti galimą riziką vartotojui;
- Siekiant sumažinti kūrimo ir palaikymo kaštus;
- Siekiant pagerinti sistemos savybes.

Išanalizavus mokslinę literatūrą apie internetinių sistemų testavimą, parengtas tarpininkavimo sistemos testavimo aprašas.

Yra penki pagrindiniai testavimo elementai, kurių laikantis vykdysime sistemos testavimo darbus (T. Kuliešius, 2008):

1. Testavimo strategija – ja naudodamiesi matysime, kokio tipo testus turėsime vykdyti, norėdami rasti programoje esančius defektus.
2. Testavimo planas – jis parodys mums, kokius testavimo darbus turėsime atlikti, norėdami laikytis mūsų nustatytos testavimo strategijos.
3. Testavimo atvejai – juos turime naudoti, siekdami patikrinti, ar programinė įranga atitinka nustatytus reikalavimus.
4. Testavimo duomenys – susideda iš įvesties duomenų ir duomenų bazėje esančios informacijos, kuri bus naudojama vykdant testus.
5. Testavimo aplinka – ji bus naudojama programinei įrangai testuoti.

Testavimo strategija – testuosime sistemą funkcinio (juodosios dėžės) testavimo stiliumi. Sudarysime testus, neatsižvelgiant į sistemos struktūrą. Testuojama sistema bus laikoma ,,dėže‘‘ su tam tikrai įėjimais ir išėjimais. Testai remsis sistemos funkciniais reikalavimais, t.y. žiniomis, kokios ,,dėžės‘‘ išėjimų vertės teisingos prie tam tikrų įėjimų (I. Plauska, 2013).

Testavimo planas:

- Išskaidyti sistemą į esminius testuojamus komponentus;
- Parengti kiekvieno komponento testavimo aprašą, kuris nusakys, ką testavimo metu mes turime pasiekti, pvz.: įrodyti, kad sukurtas atrankos algoritmas randa tinkamus meistrus;
- Parengti testavimo duomenis;
- Atlikti testavimo darbus, kiekvieną komponentą testuoti bent po keletą kartų su skirtingais duomenimis;
- Pateikti testavimo rezultatus.

Testavimo atvejai – patikrinti, kaip sukurta sistema veikia su jai pateiktais įvairiais duomenimis, kaip paieškos algoritmas reaguoja į tam tikras situacijas, kaip sistema dirba įvairiose aplinkose.

Testavimo duomenys – meistrų atrankos algoritmui ištestuoti bus naudojama Lietuvos įmonių duomenų bazė, kiti įvesties duomenys bus parinkti atsitiktinai.

Testavimo aplinka – remiantis ,,StatCounter‘‘ duomenimis, pasirinktos trys šiai dienai populiariausios interneto naršyklės: ,,Google Chrome‘‘, ,,Safari‘‘ ir ,,Mozilla Firefox‘‘, su kuriomis atliksime testavimo darbus.

Testuojant sistemą išskaidysime į šias grupes:

- Meistrų atrankos algoritmo testavimas;
- Sistemos informacinių blokų pateikimo, klaidų atsako, laiškų išsiuntimo vartotojams testavimas;
- Informacijos įvedimo, patikrinimo (angl. validation), išsaugojimo duomenų bazėje testavimas;
- Saugumo testavimas

4.4.1 Meistrų atrankos algoritmo testavimas

Pagrindinis testavimo tikslas yra įrodyti, jog sukurtas algoritmas randa tinkamus meistrus, t.y. tuos, kurie atitinka nurodytą darbų kategoriją, turi aukščiausius reitingus ir yra arčiausiai užsakovo.

Testavimo metu taip pat svarbu nustatyti, kaip algoritmas elgsis iškirtinais atvejais, neradęs tinkamų meistrų, ar radęs tik tuos, kurie yra toli nuo užsakovo arba turi mažus reitingo balus.

Testavimui atlikti į sistemos meistrų duomenų bazę buvo įkelta 50 įvairių įmonių meistrų, veiklą vykdančių Šiaulių mieste, duomenys. Meistrų duomenys paimti iš Lietuvos buitinių paslaugų teikimo įmonių duomenų bazės. Meistrų reitingų reikšmėms priskirti pseudo atsitiktiniai dydžiai, kuriuos sugeneravo sistema. Ar likę duomenys apie meistrus atitinka realybę informacijos nėra, tačiau atrankos algoritmo testavimui tai didelės įtakos neturi.

Algoritmo testavimo įvestis atsitiktinė –įvairūs Šiaulių miesto adresai. Algoritmas testuotas dešimt kartų su įvairiais adresais ir įvairiomis darbų kategorijomis.

8.Lentelė *Testavimo rezultatų duomenys.*

Įvestis			Rezultatai			
Nr.	**Adresas**	**Darbų kategorija**	**Rastų meistrų skaičius**	**Rastų meistrų adresai**	**Reitingas**	**Atstumas iki kliento**
1.	Tilžės gatvė 26, Šiauliai	Santechnika	2	Vilniaus 100, Šiauliai	4/5	3.29 km
				V. Kudirkos gatvė 25, Šiauliai	1/5	2.69 km
2.	J.Basanavičiaus gatvė 25, Šiauliai	Staliaus darbai	1	Vytauto gatvė 88, Šiauliai	3/5	2.05 km
3.	Krymo gatvė 7, Šiauliai	Elektros instaliacijos darbai	0	–	–	–
4.	V. Bielskio gatvė 71, Šiauliai	Apdailos darbai	3	Vilniaus gatvė 324, Šiauliai	5/5	6.94 km
				Gegužių gatvė 43, Šiauliai	5/5	16.48 km
				Žemaitės gatvė 20, Šiauliai	2/5	4.29 km
5.	Dubijos gatvė 17, Šiauliai	Langų keitimas	0	–	–	–
6.	Aido gatvė 12, Šiauliai	Buitinės technikos remontas	4	Dubijos gatvė 14, Šiauliai	5/5	4.23 km
				Ragainės gatvė 15, Šiauliai	5/5	3.14 km
				Žemaitės gatvė 17, Šiauliai	3/5	4.14 km
				Aušros alėja 2, Šiauliai	4/5	4.47 km
7.	Sodo gatvė 25, Šiauliai	Patalpų valymas po remonto	2	Vilniaus gatvė 213, Šiauliai	2/5	2.33 km
				Radviliškio gatvė 61, Šiauliai	4/5	5.99 km
8.	Vilkaviškio gatvė 45, Šiauliai	Namų apšiltinimas	0	–	–	–
9.	Trakų gatvė 20, Šiauliai	Kompiuterių remontas	3	Ežero gatvė 17, Šiauliai	5/5	0.85 km
				Vytauto gatvė 88, Šiauliai	3/5	0.72 km
10.	Gegužių gatvė 43, Šiauliai	Žaliuzių montuotojas	1	Darbininkų gatvė 5, Šiauliai	2/5	6.09 km

Kaip matome testavimo rezultatų duomenų lentelėje, algoritmas septyniems klientams rado meistrus, trims – ne.

Trečiai, penktai ir aštuntai užklausai meistrai nebuvo rasti, nes tokias paslaugas teikiančių įmonių / meistrų iš Šiaulių miesto sistemoje nėra.

Trečiai užklausai algoritmas rado vieną meistrą iš Šiaulių miesto, kuris teikia nurodytą paslaugą, tačiau jo reitingo balas 0. Algoritmas šį meistrą atmetė, kaip nepatikimą ir klientui nesiūlė.

Algoritmo testavimo pagrindiniai trūkumai: per maža įmonių duomenų bazė, nėra galimybės ištestuoti algoritmo veikimo, jei meistrų būtų ženkliai daugiau. Turint pakankamai didelę meistrų duomenų bazę, galima būtų pamatyti, kaip algoritmas gautus tinkamus meistrus diferencijuoja pagal atstumus iki kliento ir meistrų reitingo balus.

4.4.2 Klaidų pateikimo, laiškų išsiuntimo testavimas

Pagrindiniai šios dalies testavimo uždaviniai yra išsiaiškinti, ar sistema tinkamai pateikia visus klaidų paranešimus vartotojui, ar jie yra informatyvūs. Kitas uždavinys – išsiaiškinti, ar vartotojas gauna visus jam priklausančius laiškus, ar jų turinys pateiktas korektiškai. Laiškų siuntimas ir gavimas yra svarbi dalis sistemoje, nes per ją vyksta komunikacija su rangovais ir užsakovais.

Pirmajam uždaviniui specialiai pasirinkti nekorektiški duomenys ir mėginta juos pateikti duomenų įvestyje. Pateikimo metu buvo stebima, kokį atsaką vartotojui pateikia sistema.

9.Lentelė *Sistemos išorinės dalies vartotojo įvesties duomenų tikrinimo atsakas.*

Įvesties duomenys	**Formos laukas**	**Atsakas**
Testas.testuotojas	El. paštas	Klaidingas el. paštas
123	Įmonės kodas	Minimalus simbolių skaičius: 9
Tilžės g.	Adresas	Neteisingai įvestas adresas, nerastas miestas. Neteisingai įvestas adresas, nerasta gatvė. Neteisingai įvestas adresas, nerastas namo / pastato nr.
–	Vardas	Minimalus simbolių skaičius: 5
565256	Telefonas	Neteisingai įvestas telefono numeris (LT standartas)
„Pateikiama tuščia reikšmė“	El. paštas	Šis laukas yra privalomas
„Pateikiama tuščia reikšmė“	Įmonės kodas	Šis laukas yra privalomas
„Pateikiama tuščia reikšmė“	Adresas	Šis laukas yra privalomas
„Pateikiama tuščia reikšmė“	Vardas	Šis laukas yra privalomas
„Pateikiama tuščia reikšmė“	Telefonas	Šis laukas yra privalomas

Penkta lentelė iliustruoja, kaip vartotojas, įvedęs neteisingus duomenis, gauna klaidos atsaką. Tikrinimo atsakas yra pakankamai informatyvus, aiškiai nurodantis, ką vartotojas įvedė ne taip. Vartotojui neįvedus privalomų duomenų, sistema visur grąžina tą patį pranešimą: „Šis laukas yra privalomas“. Tai nėra visiškai informatyvu, nes konkrečiai nenurodoma, kuris laukas yra privalomas – vartotojas tai turi suprasti pagal pranešimo vietą, t.y. pagal tai, prie kurio laukelio jis yra pateiktas.

Visose tikrintose naršyklėse klaidos pranešimai pateikiami vienodai, jokių pakitimų nepastebėta.

Antrajam uždaviniui ištestuoti buvo vykdomos šeštoje lentelėje pateiktos operacijos. Jos vykdytos taip, kad laiškai būtų siunčiami, t.y. visi duomenys pateikiami teisingai – taip, kaip reikalauja sistema. Testavimo metu buvo naudotos šios elektroninio pašto tarnybos: „Gmail“, „Outlook“, „Thunderbird“.

10.Lentelė *Sistemos laiškų siuntimo testavimas*

Laiškas	**Pašto tarnyba / klientai**	**Pastebėjimai**
Vartotojo registracijos laiškas	„Gmail“, „Outlook“, „Thunderbird“	Visos pašto tarnybos laišką gavo laiku, laiško turinys buvo teisingas, atvaizdavimas nepakitęs.
Vartotojo registracijos patvirtinimo laiškas	„Gmail“, „Outlook“, „Thunderbird“	Visos pašto tarnybos laišką gavo laiku, laiško turinys buvo teisingas, atvaizdavimas nepakitęs.
Slaptažodžio priminimo laiškas	„Outlook“	„Outlook“ pašto tarnyboje slaptažodžio priminimo nuoroda neaktyvi, visose kitose tarnybose viskas veikia, kaip numatyta.
Darbo pasiūlymo laiškas	„Outlook“	„Outlook“ pašto tarnyboje darbo priėmimo nuoroda neaktyvi, visose kitose tarnybose viskas veikia, kaip numatyta.
Užsakovo duomenų laiškas	„Gmail“, „Outlook“, „Thunderbird“	Visos pašto tarnybos laišką gavo laiku, laiško turinys buvo teisingas, atvaizdavimas nepakitęs.
Meistrų įvertinimo užklausos laiškas	„Outlook“	„Outlook“ pašto tarnyboje meistro įvertinimo nuoroda neaktyvi, visose kitose tarnybose viskas veikia, kaip numatyta.

Šeštoje lentelėje matyti, kad testavimo metu rasta klaida – „Outlook‘‘ pašto tarnyboje laiške pateiktos nuorodos buvo neaktyvios, o tai pakankamai reikšminga klaida, nes būtent per pateiktas nuorodas veikia sistemos komunikavimas su meistrais ir klientais. Daugiau klaidų ar neatitikimų nebuvo rasta, visa laiškuose atvaizduojama informacija pateikta korektiškai, nė vienas laiškas į šlamštą nufiltruotas nebuvo.

Pirmame priede pavaizduotas darbo pasiūlymo meistrui laiško pavyzdys.

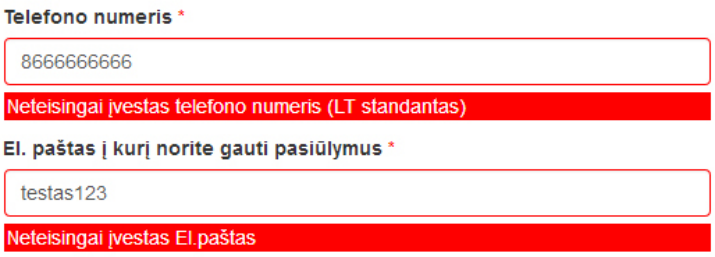
4.4.3 Informacijos įvedimo, patikrinimo, išsaugojimo duomenų bazėje, testavimas.

Informacijos įvedimo, patikrinimo (angl. validation), išsaugojimo duomenų bazėje testavimui buvo pasirinkti atsitiktiniai duomenys ir simbolių rinkiniai.

Duomenų įvedimo į sistemos formas testavime buvo atlikti šie veiksmai:

- Patikrintas duomenų įvedimas pagal numatytus standartus (pvz. el. paštas, telefono numeris ir t.t.);
- Patikrintas nekorektiškų duomenų įvedimas.

Duomenų įvedime pagal numatytus standartus patikroje klaidų neaptikta, sistema įvesties tikrinimą atlieka pagal numatytas taisykles. Vienas iš duomenų įvesties patikros pavyzdžių pateiktas dešimtame paveikslėlyje.



12 pav. **Duomenų įvedimo pagal numatytus standartus patikra**
Šaltinis: Sudaryta autoriaus

Nekorektiškų simbolių įvestyje mėginta vesti įvairius simbolių rinkinius, galinčius pakenti duomenų bazei. Sistema šiuos simbolius priėmė, tačiau juos išsaugojo, kaip tekstinę reikšmę, o tai duomenų bazės veikimui jokios įtakos neturi. Už šios dalies

patikrą yra atsakingas pasirinktas PHP karkasas, kuris jau yra ištestuotas programuotojų bendruomenės, todėl į didesnius šios dalies testavimo darbus išsiplėsta nebuvo.

Duomenų bazės patikrinime buvo atlikti šie darbai:

- Patikrintas duomenų vientisumas, kaip yra trinamos, keičiamos formos ar atliekamos kitos su duomenų baze susijusios funkcijos;
- Patikrinta, ar visos užklausos įvykdomos teisingai, ar gaunami teisingi duomenys.

Duomenų bazės vientisumui patikrinti buvo atliekami įvairūs veiksmai sistemoje: vykdoma meistro registracija, koreguojami meistrų duomenys, pateikiamas darbo pasiūlymas ir t.t. Taip pat buvo stebimas informacijos atsinaujinimas duomenų bazėje. Stebėjimas vyko per interneto naršyklę, prisijungus prie duomenų bazės valdymo platformos. Pagrindinis dėmesys buvo skirtas duomenų vientisumui. Pavyzdžiui, analizuota, ar ištrynus bazinę meistro informaciją iš pagrindinės lentelės, ji dingsta ir iš kitų lentelių. Stebėjime nuokrypių pastebėta nebuvo. Klaidų tikimybė buvo ganėtinai maža, nes sistemos duomenų bazės projektas su sąryšiais tarp lentelių sukurtas ,,MySQL Workbench‘‘ programa, kurios pagalba buvo suprojektuota duomenų bazės projekto schema ir išeksportuota, kaip Sql kodas. Pateikus šį kodą į MySQL duomenų bazės platformą, susikūrė visos numatytos lentelės ir jų sąryšiai. Kuriamo projekto duomenų bazės schema pateikta antrajame priede.

4.4.4 Saugumo testavimas

Testuojant sistemos saugumą buvo atlikti šie darbai:

- Testuota neprisijungus: ar atsidaro vidinis puslapis, įterpus vidinį universalųjį adresą (URL) tiesiai į naršyklės adreso juostą.
- Testuota, kokia sistemos reakcija po neteisingų įvedimų (pvz., vartotojo vardo, slaptažodžio).
- Testuota, ar leidžiama prieiga, jei prisijungus tam tikru vartotojo vardu bei slaptažodžiu ir naršant po sistemos vidinius puslapius, tiesiogiai pakeičiamas URL, pvz. saito ID į kito saito ID, kuris nėra susijęs su prisijungusiu vartotoju.

- Testuojant saugumą buvo atsižvelgta tik į sukurtą sistemos dalykinę sritį. Globalūs PHP kodo saugumo aspektai nepaliesti, nes tarpininkavimo sistema buvo programuojama ant jau sukurto, kūrėjų ir programuotojų bendruomenės ištestuoto PHP karkaso.

11.Lentelė *Sistemos saugumo testavimas*.

Testuojama dalis	**Veiksmas**	**Atsakas**
Testavimas neprisijungus	Neprisijungus mėginta eiti i meistro profilio kortelę su nuoroda: http://kvieciumeistra.lt/meistru–katalogas/profilis	Nukreipia į prisijungimą prie paskyros: http://kvieciumeistra.lt/prisijungimas
Sistemos reakcija po neteisingai įvedamų prisijungimo duomenų	Mėginta į prisijungimo prie paskyros formą vesti neteisingus prisijungimo duomenis.	Prisijungimo laukai paraudonuoja, forma keletą kartų sujuda (įvyksta animacija, skirta atkreipti vartotojo dėmesį), puslapis nepersikrauna.
Testavimas, ar suteikiama prieiga prie svetimų duomenų	Prisijungus prie vartotojo paskyros ir įėjus į vartotojo duomenų redagavimą, nuoroda: http://kvieciumeistra.lt/meistru–katalogas/duomenu–redagavimas/34 buvo pakeistas paskutinis skaičius, kuris nusako vartotojo ID.	Pakeitus paskutinį skaičių iš 34 į 33 (kito, egzistuojančio vartotojo ID) sistema atliko nukreipimą atgal į prisijungusio vartotojo profilio paskyrą, nuoroda: http://kvieciumeistra.lt/meistru–katalogas/profilis

Iš septintos lentelės matyti, kad sistemoje saugumo klaidų nebuvo aptikta.

Apibendrinant testavimo rezultatus, galima teigti, kad buvo rasta tik viena esminė klaida – „ Outlook‘‘ pašto tarnyboje pateiktos sistemos nuorodos neaktyvios. Ištaisius šią klaidą, sistema galima būtų leisti naudotis vartotojams, prieš tai juos informavus, kad tai pradinė – pilotinė sistemos versija.

Paleidus sistemą reikia stebėti meistrų registraciją, o užsiregistravus didesniam meistrų skaičiui, pavyzdžiui, 500, grįžti prie atrankos algoritmo testavimo, peržiūrėti, kaip algoritmas susitvarko su didesniu atrankos duomenų kiekiu.

IŠVADOS

1. Išanalizavus tarpininkavimo sistemas, nustatyta, kad svarbiausias jų komponentas – atrankos mechanizmas, užtikrinantis sistemų darbo efektyvumą ir konkurencingumą rinkoje. Analizuotų sistemų atrankos mechanizmai sudaryti iš dviejų dalių: atranka pagal miestą ir atranka pagal darbų kategoriją. Toks atrankos modelis yra supaprastintas, todėl nėra pakankamai efektyvus.

2. Technologinių sprendimų lyginamoji analizė parodė, kad populiariausios ir plačiausiai naudojamos web programavimo kalbos yra Java, PHP, Python, ir ASP.net. Pranašiausia ir tinkamiausia kuriamam projektui realizuoti PHP programavimo kalba. Projekto kūrimo darbų optimizavimui rekomenduojama naudoti programavimo karkasą (angl. framework). Vienas iš pažangiausių ir populiariausių PHP programavimo karkasų yra „Laravel‘‘.

3. Suprojektuotas kuriamos sistemos modelis su unikaliu rangovų atrankos algoritmu, atrenkančiu meistrus, esančius arčiausiai užsakovo ir turinčius aukščiausią patikimumo balą, skaičiuojamą automatiškai, įvertinant rangovų atliktą darbą.

4. Sistemos testavimui pasirinktas funkcinis (juodosios dėžės) testavimo stilius, pagal T. Kuliešių pritaikyti penki pagrindiniai testavimo elementai, sudarytas testavimo planas, kuriuo remiantis nustatyta, kad:

4.1. Meistrų atrankos algoritmas veikia tinkamai, t. y. atsižvelgia į atstumą iki kliento ir meistro reitingą (50 meistrų su skirtingais reitingais ir paslaugų tipais);

4.2. Klaidų pateikimo ir laiškų išsiuntimo testavimas su skirtingomis programomis parodė, kad sistema visose testuotose programose nurodo vartotojui klaidas ir laiškuose tinkamai atvaizduoja informaciją;

4.3. Informacijos įvedimo, patikrinimo ir išsaugojimo duomenų bazėje testavimas parodė, kad sistema užtikrina informacijos vientisumą bei apsaugo vartotoją nuo nekorektiškų duomenų įrašymo;

4.4. Sistema yra saugi, nes atpažįsta netinkamus URL, reaguoja į netinkamai įrašytus vartotojų identifikavimo duomenis, o prisijungus ir naršant po sistemos vidinius puslapius, tiesiogiai pakeitus URL, sistema blokuoja prieigą.

LITERATŪRA

1. Mahanata, A. (2005*). PHP, MySQL ir ASP.NET, Web Hosting.* Žiūrėta 2015, gruodžio 5 d. internete: < http://www.articlesphere.com/lt/Article/PHP—MySQL–and–ASP–NET–in–Web–Hosting/122878>.

2. Informacinės visuomenės plėtros komitetas (2013) *Informacinės visuomenės plėtros 2013 metų apžvalga.*

3. Išmanusis telefonas lietuviams tėra papuošalas (2012). Žiūrėta 2015, gruodžio 7 d. internete: < http://ekonomika.tv3.lt/naujiena/ismanusis–telefonas–lietuviams–tera–papuosalas–24321.html>.

4. *TIOBE Index for December* (2015). Žiūrėta 2015, gruodžio 7 d. internete: < http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.

5. Kohan, B. (2010). *PHP vs ASP.net Comparison.* Žiūrėta 2015, gruodžio 10 d. internete: < http://www.comentum.com/php–vs–asp.net–comparison.html>.

6. *PHP frameworks comparison* (2015). Žiūrėta 2015, gruodžio 9 d. internete: < http://socialcompare.com/en/comparison/php–frameworks–comparison>.

7. *The Best PHP Framework for 2015: SitePoint Survey Results* (2015). Žiūrėta 2015, gruodžio 10 d., internete: < http://www.sitepoint.com/best–php–framework–2015–sitepoint–survey–results/>.

8. *System Properties Comparison Microsoft SQL Server vs. MySQL vs. Oracle* (2015). Žiūrėta 2015, gruodžio 11 d. internete: < http://db–engines.com/en/system/Microsoft+SQL+Server%3BMySQL%3BOracle>

9. *What Programming Language Should You Learn?* (2016). Žiūrėta 2016, kovo 15 d. internete: < https://www.learneroo.com/modules/9/nodes/620>.

10. *A small place to discover languages in GitHub* (2016). Žiūrėta 2016, kovo 29 d. internete: < http://githut.info/>.

11. Woler, R. (2012). *Living Apart Together: Decoupling Code and Framework.* Žiūrėta 2016, kovo 10 d. internete: < http://www.sitepoint.com/living–apart–together–decoupling–code–and–framework/>.

12. Vaitkūnas, M. (2013). *PHP karkasų architektūrinis tyrimas.* Žiūrėta 2016, kovo 19 d. internete: < http://vddb.library.lt/fedora/get/LT–eLABa–0001:E.02~2013~D_20130826_102846–07165/DS.005.0.01.ETD/>.

13. Oleškevič, M. (2014). *Programavimo kalbų naudojimo Lietuvos mokyklose analizė.* Žiūrėta 2016, kovo 23 d. internete : < http://vddb.library.lt/fedora/get/LT–eLABa–0001:E.02~2014~D_20140723_101945–13824/DS.005.0.01.ETD/>.

14. Alona Z. (2016). *TOP 20: Programming Languages Research–201.* Žiūrėta 2016, kovo 30 d. internete : < https://www.cleveroad.com/blog/top–20–programming–languages–research–2016/>.

15. R. W. Sebasta (2008). *Concepts of Programming Languages.*

16. D., Dzemydienė, G., Romeika, M., Okulič Kazarinas. (2016). *Elektroninių tarpininkų funkcijos ir galimybės teikti paslaugas statybos ir remonto srityje.* Žiūrėta 2016, kovo 8 d. internete: < https://www.researchgate.net/profile/Mykolas_Okulic–Kazarinas/publication/282122910_Elektroniniu_tarpininku_funkcijos_ir_galimybes_teikti_paslaugas_statybos_ir_remonto_srityje/links/5603bb3608ae460e2704f4f1.pdf/>.

17. Etalinq. (2013). *Tarpininkavimo verslas internete.* Žiūrėta 2016, kovo 3 d. internete: < http://etalinq.com/tarpininkavimo–verslas–internete/>.

18. J.,Katina. (2009). *Mobilių aplikacijų projektavimo technologijų analizė.* Žiūrėta 2016, kovo 10 d. internete : < http://vddb.library.lt/fedora/get/LT–eLABa–0001:E.02~2009~D_20090626_094812–29762/DS.005.0.02.ETD/>.

19. E.,Matijošius. (2009). *Žiniatinklinių tinklalapių ir DBVS analizavimas bei optimizavimas.* Žiūrėta 2016, kovo 11 d. internete: < http://vddb.library.lt/fedora/get/LT–eLABa–0001:E.02~2008~D_20090908_201756–82885/DS.005.1.01.ETD/>.

20. A., Berger. (2013). *The basics of embedded software testing: Part 1.* Žiūrėta 2017, balandžio 3 d. internete: <http://www.embedded.com/design/other/4212929/4/The–basics–of–embedded–software–testing—Part–1/>.

21. I., Plauska. (2013). *Įterptinių sistemų testavimas.* Žiūrėta 2017, balandžio 3 d. internete: <http://kopustas.elen.ktu.lt/studentai/lib/exe/fetch.php?media=embedded_testing_ignas_plauska.pdf/>.

22. *Browser Market Share Worldwide* (2017). Žiūrėta 2017, kovo 4 d. internete: < http://gs.statcounter.com/>.

23. *Šiuolaikinių duomenų bazių programavimo metodų tyrimas* (2006). Žiūrėta 2017, balandžio 24 d. internete: <http://talpykla.elaba.lt/elaba–fedora/objects/elaba:2086174/datastreams/MAIN/content>.

24. *DB – Engines Ranking* (2017). Žiūrėta 2017, balandžio 24 d. internete: < https://db–engines.com/en/ranking>.

ANOTACIJA

Autorius: Aivaras Karnatka

Tema: *Buitinių paslaugų tarpininkavimo sistemos modelio tyrimas.*

Šiaulių universitetas 2017.

Informacinėms technologijoms vis sparčiau skverbiantis į žmonių gyvenimą, auga poreikis kurti taikomąsias programas, skirtas kasdieniams darbams palengvinti ar atlikti įvairias užduotis, pavyzdžiui, išsikviesti taksi.

Buitinių paslaugų sfera – taip pat ne išimtis. Joje vis svarbesnį vaidmenį atlieka elektroninės tarpininkavimo sistemos, kurios padeda užsakovams nepasimesti skelbimų gausoje, ieškant tinkamo meistro, auklės, valytojo, kitas buitines paslaugas teikiančio asmens ar įmonės. Deja, šiandien Lietuvoje tokio pobūdžio tarpininkavimo sistemų, kurios būtų patogios įvairiems vartotojų tipams, rastų tik tinkamus ir patikimiausius meistrus, veiktų nemokamai – nėra.

Šio darbo tikslas – suprojektuoti buitinių paslaugų tarpininkavimo sistemos modelį ir jį ištirti. Įgyvendinat projektą buvo sukurtas buitinių paslaugų tarpininkavimo sistemos prototipas, kuriame realizuotas unikalus rangovų atrankos mechanizmas.

ANNOTATION

Author: Aivaras Karnatka

Subject: *Domestic service mediation system model study.*

Šiauliai University, 2017.

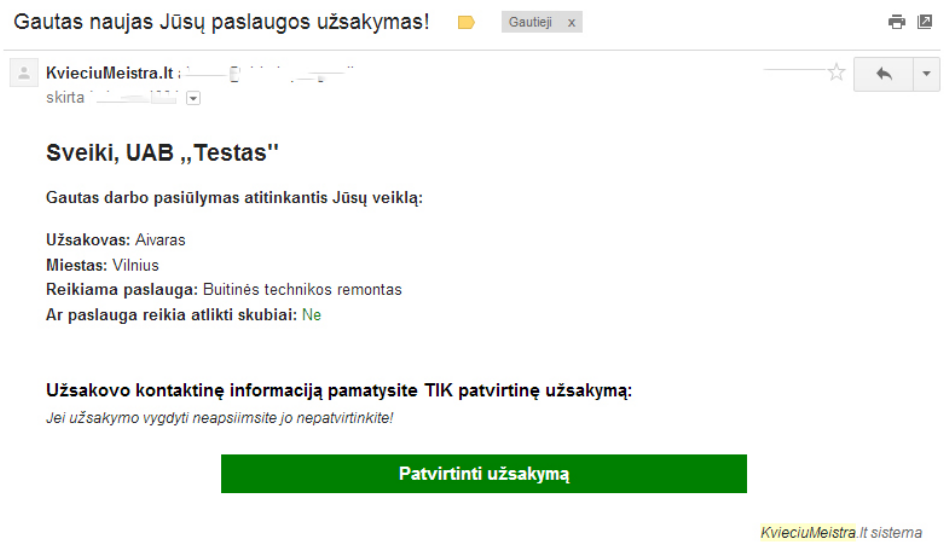
Rapid penetration of information technologies into the daily lives of people is associated with a growing need for development of applications designed for facilitation of daily chores or carrying out various tasks, like calling a taxi.

The field of household services is no exception along with increasing significance associated with the role of electronic intermediation systems, which help customers not to get lost in abundance of advertisements encountered within the course of search for the right handyman, babysitter, cleaner, or other person or company engaged in provision of

household services. Unfortunately, as for today no such intermediation systems of this type, which would be convenient for various types of users, would find only competent and reliable handymen, are available for operation in Lithuania free of charge.

The purpose of this Project is to design a model of household services intermediation system and to analyse it. A prototype of household services intermediation system was developed within the course of implementation of the project realizing a unique contractor screening mechanism.

1 PRIEDAS



Gautas naujas Jūsų paslaugos užsakymas! Gautieji x

KvieciuMeistra.lt
skirta

Sveiki, UAB „Testas"

Gautas darbo pasiūlymas atitinkantis Jūsų veiklą:

Užsakovas: Aivaras
Miestas: Vilnius
Reikiama paslauga: Buitinės technikos remontas
Ar paslauga reikia atlikti skubiai: Ne

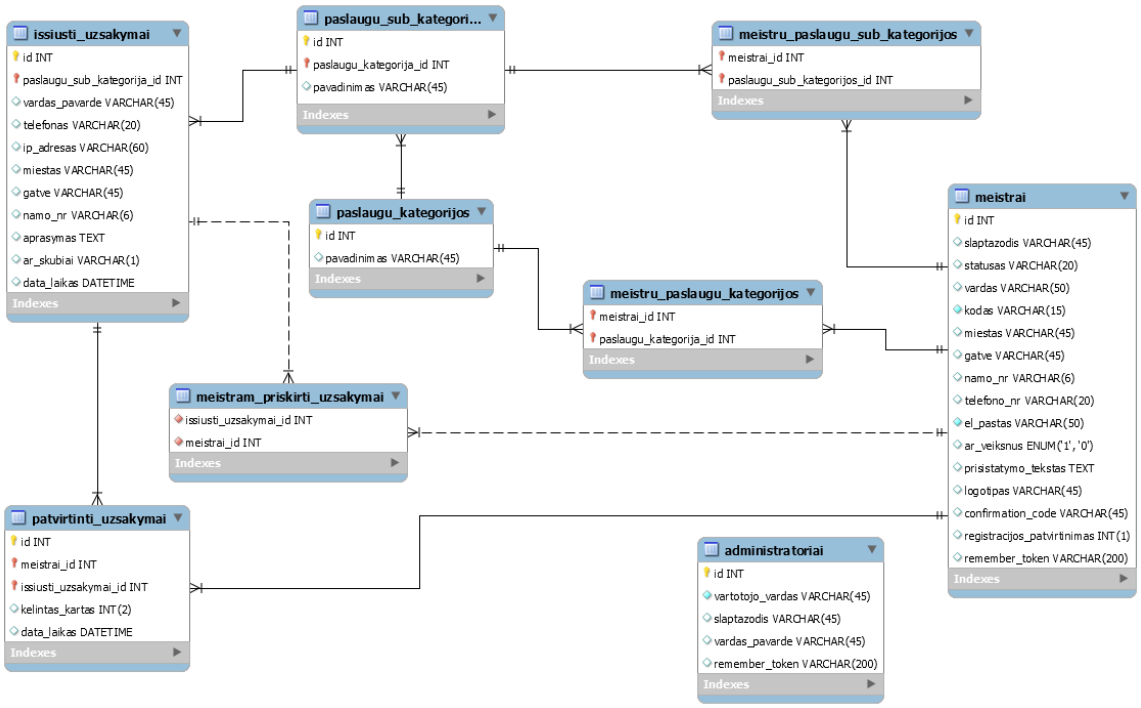
Užsakovo kontaktinę informaciją pamatysite TIK patvirtinę užsakymą:
Jei užsakymo vygdyti neapsiimsite jo nepatvirtinkite!

Patvirtinti užsakymą

KvieciuMeistra.lt sistema

Darbo pasiūlymo pateikimas rangovui.

2 PRIEDAS



Kuriamos sistemos duomenų bazės projektas.