Taikomieji moksliniai tyrimai / Applied Scientific Research ISSN 2783-6290
2024, 3(1), 7-20. DOI: https://doi.org/10.56131/tmt.2024.3.1.159 Gauta 2023-07-20. Publikuota 2024-01-13

Ekspertinių sistemų kūrimo įrankio/platformos projektavimas

Donatas Daugirdas
Šiaulių valstybinė kolegija, Informatikos mokslų katedros lektorius
Šiaulių valstybinė kolegija / Higher Education Institution, Lithuania; Lecturer at the Department of Informatics Sciences
d.daugirdas@svako.lt

Karolis Paliliūnas
Šiaulių valstybinė kolegija, Programų sistemų studijų programos absolventas
Šiaulių valstybinė kolegija / Higher Education Institution, Lithuania; Graduate of the Program System study program

Anotacija

Straipsnyje analizuojamas įmonės poreikis ekspertinių sistemų kūrimo prototipo sukūrimui. Išnagrinėtos ekspertinės sistemos ir jų kūrimo įrankiai, nustatytos įmonės poreikį atitinkančios funkcijos ir programinė įranga ekspertinės sistemos modelio sukūrimui. Aprašyti ekspertinės sistemos projektavimo ir programavimo procesai, sistemos veikimo testavimas ir sistemos dokumentacijos parengimas, sistemos įdiegimas į serverį.

Reikšminiai žodžiai: ekspertinės sistemos, ekspertinių sistemų kūrimo įrankiai, projektavimas.

Expert System Development Prototype Project

Summary

The article analyzes the company's need to create a prototype for the development of expert systems. Expert systems and tools for their creation were examined, functions and software for the creation of an expert system that met the needs of the company were determined. The expert system design and programming processes, system performance testing and system documentation preparation, installation of the system on the server are described.

Keywords: expert systems, expert systems development tools, design.

Įvadas

Projekto aktualumas. Ekspertinė sistema yra dirbtinio intelekto elementas, kuris sugeba imituoti eksperto veiklą ir naudoti iš anksto apibrėžtas žinių bazes bei samprotavimo grandines, jų pagalba išvesdamas galutinį variantą [8]. Pagrindinis ekspertinių sistemų privalumas yra tai, kad jos suteikia ekspertams įrankį, kaip pateikti savo vertinimą ar problemos sprendimo apibendrinimą. Jos gali būti ilgos ir sudėtingos, turėdamos šimtus ar tūkstančius galimų baigčių, pagrindinių ir papildomų klausimų, suteikia galimybę daugeliui vartotojų gauti eksperto sprendimus ir atsakymus, be tiesioginio vartotojų dalyvavimo.

Kuriant ekspertines sistemas, kurios programinės sistemos, galinčios atlikti specialistų darbą arba spręsti sudėtingus uždavinius tam tikroje srityje, galima naudoti įvairius technologinius sprendimus. Ekspertinių sistemų kūrimo įrankiai leidžia naudotojams sukurti pagal poreikį savo specializuotą klausimyną su pilna samprotavimo grandine ir pateikti kitiems naudotojams jį įvykdyti, taip sukuriant galimybę imituoti eksperto vertintą pasirinkimų grandinę ir pateikti išvadą pagal iš anksto aprašytas taisykles. Sukurta detalizuota sistema yra įrankis tokio klausimyno įgyvendinimui. Ekspertinių sistemų kūrimo įrankis leidžia vartotojams sukurti klausimynus apie specifinę problemą, apie bet kokio prietaiso, įrangos ar kitos sistemos panaudojimo galimybes žinių savikontrolei ir duomenų analitikai [1]. Šiame straipsnyje apibūdinta sistema, sukurta pagal įmonės UAB „Gamybos kodo technologijos“ poreikį.

Pagrindinis straipsnyje aprašomas ekspertinių sistemų tikslas yra sukurti ekspertinių sistemų kūrimo įrankį, kuris turėtų visus samprotavimo grandinės realizavimo įrankius su papildomais dirbtinio intelekto elementais, kurie leidžia naudotojams sukurti ekspertinę sistemą, pasidalinti ja su

naudotojų grupėmis ar kitais naudotojais. Toks modelio kūrimo principas leidžia įvairių sričių ekspertams lengvai ir greitai pateikti sudėtingų sprendimų klausimyną, kuris tam tikru keliu nuveda asmenį, atliekantį ekspertinės modelio pasirinkimus, prie jo samprotavimo grandinės išvados.

Projekto problema – įmonė UAB „Gamybos kodo technologijos“ išreiškė poreikį ekspertinės sistemos modelio kūrimo įrankiui. Tokių technologinių sprendimų sistemų atviro kodo pasaulyje yra nedaug (pavyko rasti 2 atviro kodo ekspertines sistemas), o lietuviškų sistemų iš viso nėra, todėl yra keliama užduotis sukurti tokio tipo sistemą vadovaujantis įmonės poreikiais.

Projekto objektas – ekspertinių sistemų kūrimo įrankis.

Projekto tikslas – suprojektuoti ir sukurti ekspertinių sistemų kūrimo, administravimo ir vykdymo sistemą.

Projekto uždaviniai:

1. Išanalizuoti kitas ekspertines sistemas ir jų kūrimo įrankius, nustatyti tinkamas funkcijas ir jas realizuoti projekte.
2. Pasirinkti geriausiai tinkančią programinę įrangą sistemos kūrimui, išanalizavus jos poreikius.
3. Suprojektuoti sistemos modelį, realizuojant numatytąsias funkcijas.

Projekto metodai: panašių sistemų analizė, mokslinės literatūros analizė, unikalaus projekto prototipo kūrimas.

Ekspertinės sistemos kitaip žiniomis grindžiama sistema

Ekspertinė sistema (toliau – ES) yra kompiuterizuota programa, kuri veikia kaip ekspertas tam tikroje srityje. Ši sistema atlieka tam tikrą užduotį, pavyzdžiui, padeda išanalizuoti kylančias problemas darbo vietoje prie staklių, skaičiuoja, analizuoja reikalingą informaciją, prognozuoja išvadas ir pateikia sprendimus panašiai, kaip tai padarytų žmogus su geromis žiniomis konkrečioje srityje. Ekspertinė sistema naudoja „žinių bazių valdymo principus“, kuriuos taiko žmogus sprendžiant tam tikras problemų kategorijas. Tai leidžia ekspertinėms sistemoms įgyti ekspertų patirties ir žinių, modeliuojant jų sprendimo procesą. Šio tipo programų tikslas yra padidinti darbo efektyvumą ir sutaupyti laiką, kai reikia gauti sprendimus, kuriems reikia daug žinių ir patirties [10]. ES nustato duomenų struktūrą bei pritaiko sprendimo modelius, kuriuose yra įtrauktos žmogaus patirties žinios. Veikiant ekspertinei sistemai, procesas apima du etapus: pirmasis yra žinių bazės užpildymas, o antrasis yra problemos, kuri turi būti išspręsta, pateikimas modelio veikimo principams. Ekspertinė sistema gauna vartotojo įvestį ir pagal jo pasirinkimų šabloną iš žinių bazės ištraukia atitinkamą taisyklę arba išvadą. Modelio patikimumą ir veikimo kokybę lemia žinių bazės vertinimo kokybė, tinkamas bei kompetentingas veikimo organizavimas [9].

Ekspertinė sistema yra dirbtinio intelekto šaka, naudojanti specifines žinias, padedančias priimti konkrečios modelio srities ekspertinius sprendimus, atitinkančius žmogaus eksperto sprendimus. Ekspertinių sistemų valdymo aplinka iš esmės yra įrankis parengti testą, klausimyną, sprendimo paiešką iš aktualių temų naudotojui. Ekspertinės sistemos gali pagal pateiktą pilną samprotavimo grandinę ir pagal dirbtinio intelekto elementų metodiką atvesti sistemos naudotoją prie tikslaus pasirinkimo [11]. Šių tipų sistemos atlieka skirtingas funkcijas, tačiau, pasitelkus klausimyno redaktoriaus galimybes ir dirbtinio intelekto elementus, galima suprojektuoti ir sukurti ekspertinių sistemų valdymo aplinką su integruotomis papildomomis funkcijomis bei pritaikyti tokią sistemą pagal užsakovo reikalavimus. Tradicinių ekspertinių sistemų sukūrimas, imituojant ekspertų samprotavimo modelį ir su juo susijusį žinių įgijimo procesą, yra sudėtingas uždavinys.

Vadovaujantis moksliniais šaltiniais, žemiau išnagrinėtos ekspertinės sistemos ir jų redaktoriai, apibūdinta, kaip sistemos veikia, koks jų panaudojimas ir kaip jos sprendžia problemas. Aprašytos ekspertinių sistemų ir jų valdymo aplinkos, jų veikimo logika, ką jos atlieka, pateikti praktiniai pavyzdžiai ir panašių sistemų vertinimai.

Ekspertinės sistemos modelio kūrimo teorinis diskursas

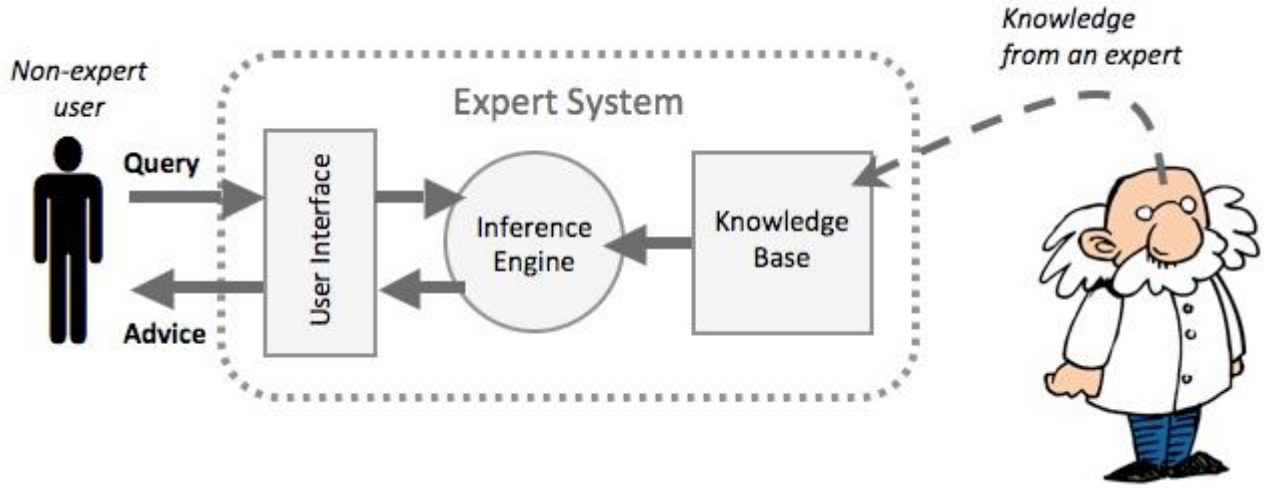
Ekspertinė sistema (toliau – ES) imituoja žmogaus eksperto sprendimų priėmimo gebėjimą. ES žinios gali būti ekspertinės arba bendrosios. Šiuolaikinės ekspertinės sistemos veikia pagal nustatytas

funkcionalumo ribas. ES kuriamos aiškiai apibrėžtos srities problemoms spręsti. Kaip ir žmogus ekspertas, ekspertinė sistema veikia siauroje kompetencijų srityje, už kurios ribų sistema nepajėgia priimti tinkamų sprendimų (pvz., ekspertinė medicinos sistema nebus naudinga sprendžiant statybos reglamentavimo problemas). Sistemos žinios apie specifinių problemų sprendimą vadinamos ES žinių sritimi.

Bendroji koncepcija, pasiūlyta jos pradininko prof. E. Feigenbaumo, ES apibrėžia kaip „*išmaniąją kompiuterinę programą, naudojančią žinias ir išvadų generavimo procedūras problemoms, kurioms reikia gerų tos srities specialistų kompetencijų, spręsti*“ [2]. Tokio tipo ekspertinės sistemos vadinamos „*taisyklėmis pagrįstomis ekspertinėmis sistemomis*“. Toks tipas yra dažniausia ir populiariausia ekspertinės sistemos rūšis.

Pirmoji sukurta ekspertinė sistema buvo „MYCIN“, Stanfordo universitete 1970-aisiais. Ji buvo skirta nustatyti kraujo infekcijas ir rekomenduojamus gydymus. Tokio modelio sistema tapo labai sėkminga ir išpopuliarino ekspertinės sistemos principą pasaulyje [4]. Ekspertinių sistemų technologija yra mechanizmas, leidžiantis kurti kompanijos ar firmos žinių banką. Ekspertinės sistemos įmonėje padeda išsaugoti ir dokumentuoti ten dirbančių ekspertų žinias ir įgytas patirtis, kad jos nebūtų prarastos, ekspertams pasišalinus iš darbo vietos. Todėl ekspertinių sistemų žinių bazių panaudojimas skirtingose veiklos srityse tapo sėkminga investicija daugeliui tyrinėtojų, net ir pritaikant jas eksperimentiniu būdu. Yra keli šimtai, gal net ir tūkstančiai ekspertinių sistemų tobulinimo ir vystymo studijų įvairioms sritims, tokioms kaip medicina, kariuomenė, chemija, inžinerija, gamyba, įmonės valdymas ir kt.

Susidomėjimas ekspertinių sistemų ir dirbtinio intelekto (toliau – DI) mokslais stabiliai didėja. Tai rodo daug įvairių tyrimų, darytų 1996-2012 metais [28]. Šie tyrimai skatina aktyviai įsitraukti į DI vystymą įvairiose srityse. DI vystymas, ekspertinių sistemų kūrimas tampa vis populiaresne ir naudingesne mokslo sritimi. Vienas iš pagrindinių veiksnių, lėmęs ekspertinių sistemų populiarumo šuolį, yra jų suteikiamos galimybės pateikti sprendimus įmonėms, kurios turi sunkumų greitai besikeičiančioje pasaulinėje rinkoje. Jos padeda įmonėms greitai optimizuoti savo sprendimus ar produktus, pritaikant laiko patikrintas žinių bazes, ir apmokyti savo dirbantį personalą [7].



1 pav. Ekspertinės sistemos paprastas modelis [5]

Ekspertinėms sistemoms kurti yra naudojamos ekspertinių sistemų kūrimo aplinkos (angl. *Expert system shell*), kurios leidžia bet kuriam naudotojui sukurti ekspertinės modelio modelį kaip atskirą savarankišką produktą arba kaip kažkokios modelio plėtinį, priklausomai nuo naudojamos programavimo kalbos[28]. Jei tai yra integruota į kitą modelio programą, joms sukurti reikalingos papildomos programavimo žinios. Pavyzdžiui, statistikos profesorius gali sukurti savo studentams ekspertinę sistemą su savo paruošta medžiaga ir klausimais. Tai bus ekspertinė sistema, nes, norėdamas gauti teisingą atsakymą, studentas privalo turėti statistikos žinių ir žinoti statistikos terminus bei jų apibrėžimus. Be profesoriaus medžiagos ir žinių nekvalifikuotas žmogus negali paruošti tokios modelio . Mokymui gali būti naudojamos dvejų tipų ekspertinės sistemos:

1) Instruktoriaus pateiktos ekspertinės sistemos: instruktorius gali naudoti ekspertinių sistemų aplinką, sukurti ir pateikti „patarėją“ studentams, kur instruktorius gali sudėlioti savo mąstymą ir mintis į pasirinkimų modelį, suteikdamas studentams savo žinių banką užduoties forma;

2) Studentų pateiktos ekspertinės sistemos: tai būtų studentų sukurtos ekspertinės sistemos, naudojant kūrimo aplinką. Norint sukurti tokią sistemą, studentai turi susipažinti su savo subjektu, suprasti ir pritaikyti žinių bazės principą kompiuterinėje erdvėje. Studentai renka žinias per semestrą ir paskui atvaizduoja jas ekspertinės sistemos moduliu [3]. Turint ekspertinės sistemos apibrėžimą ir panaudojimo pavyzdžių, galima susidaryti bendrą vaizdą apie ekspertines sistemas ir jų praktinį panaudojimą.

1 lentelė

Ekspertinių sistemų ir jų pritaikymo pavyzdžiai [5]

Nr.	Kategorija	Tinkamumas problemos sprendimui	Taikymo programos (ES pavyzdžiai)	Nuorodos į šaltinius
1	Diagnozė	Rekomenduoja vaistus / įvairių ligų gydymą	SHYSTER-MMYCIN, ONCONCIN, DExS	O‘ Collagham, 2003; Saritas et al., 2013
2	Tvarkymas	Siūlo suplanuotą tvarkaraštį ir struktūrą	ESPCRM	Leng ir Teng, 1992
3	Instrukcija	Mokymo programa įvertinti vartotojo gebėjimus panaudoti turimas žinias	HSPEXP	Liebowitz, 1995
4	Interpretacija	Analizuoja įvestis, sprendžia apie jų svarbą (lyginant su žinių baze)	Hepaxpert	Rudgier et al., 2010
5	Spėjimas	Atspėja arba sudaro nuomonę apie galimą rezultatą	DEREK, stAR	Syed Abdullah et al., 2011
6	Numatymas	Numato degalų kainos pasikeitimus	GMDH tinklo integracija su „Rile“ tipo ekspertine sistema	Abrishami ir Varahrami, 2012
7	Dizainas ir planavimas	Kuria ir sumodeliuoja sprendimą greituoju būdu, keliais kartais greičiau nei žmonės ekspertai	COMEX, CAKES-ISTS	Grahivac ir Devedzic, 2010; Lee ir Lee, 2012
8	Stebėjimas	Lygina pastebėjimus, kad būtų surastos silpnosios vietos	VES	Ebersbach ir Peng, 2008
9	Kontrolė	Interpretuoja, nuspėja, taiso ir stebi modelio veikimą	KBSs	Staley, 1991
10	Stebėjimo kontrolė	Stebi atliekamus sprendimus, svarbius modelio procesams	Struxure Ware Power Monitoring Expert 7	Moridis et al., 2013
11	Klasifikacijos identifikavimas	Klasifikuoja, atpažįsta modelio tikslus pagal jos funkcijas ir gebėjimus	DENDROID	Tangil et al., 2014
12	Atradimas	Padeda vartotojui pasiruošti ir naršyti po sistemą	SeTES	Moridis et al., 2013
13	Derinimas	Pasiūlo sprendimus palaipsniui spręsti sudėtingas problemas	SIPDES	Doukidis ir Paul, 1991
14	Pasirinkimas	Pasirenka tinkamą įrankį	CNC	Tan et al., 2012

Projekto poreikių analizė

Įvairios dirbtinio intelekto technologijos yra esminiai ekspertinių sistemų kūrimo komponentai, leidžiantys sistemos suprasti ir taikyti kompleksinius sprendimus remiantis turimais duomenimis. Nauji technologiniai sprendimai dažnai apima programinės įrangos kūrimą, dirbtinį intelektą, duomenų analizę ir kitus komponentus. Keletas galimų technologinių sprendimų ekspertinių sistemų kūrimui:

1. Dirbtinis intelektas (DI):

• Mašininis mokymas: Ši technologija leidžia sistemai išmokti iš duomenų, tobulėti ir prisitaikyti prie kintančių sąlygų. Gali būti naudojami įvairūs mašininio mokymo algoritmai, pvz., neuroniniai tinklai, k-vidurkių metodai, sprendimų medžiai ir kt.

• Gilusis mokymasis: Gilių neuroninių tinklų naudojimas gali padėti modeliuoti sudėtingas žmogiškas žinias ir savybes.

2. Sisteminės inžinerijos metodologijos:

• Formalioji specifikacija: Tai leidžia detalizuoti sistemos reikalavimus ir elgesį, padeda išvengti neapibrėžtumo ir užtikrina aukštą sistemos kokybę.

• Prototipavimas: Greitas prototipų kūrimas gali būti naudingas, siekiant suprasti sistemos funkcionalumą.

3. Duomenų valdymas:

• Duomenų bazių valdymas: Efektyvi duomenų bazė gali leisti sistemos komponentams saugoti ir naudoti informaciją greitai bei efektyviai.

4. Programinės įrangos kūrimo platformos/įrankiai:

• Specializuotos programinės įrangos kūrimo aplinkos: Tokios aplinkos gali optimizuoti ekspertinių sistemų kūrimo procesą ir suteikti programuotojams efektyvius įrankius.

Šie technologiniai sprendimai gali būti taikomi kartu, priklausomai nuo konkrečios ekspertinės sistemos tikslų ir reikalavimų. Svarbu atsižvelgti į užsakovo specifiką ir užduotis, kurioms reikia sprendimo, siekiant pasirinkti tinkamiausius įrankius ir technologijas.

Įmonėje UAB „Gamybos kodo technologijos" susidurta su problema – nebuvo jokios sistemos naujų darbuotojų mokymui. Nauji darbuotojai, atėję dirbti į įmonę, susipažinimui su įmonės viduje esančia programuotojų ekosistemą, atlikdavo užduotis, kurias vadovas turėjo pats paruošti ir kiekvieną kartą pritaikyti priklausomai nuo būsimojo darbuotojo patirties. Nebuvo modelio/sistemos, kurioje būtų galima pagal darbuotojo atliktus sprendimus nuspręsti, kokią poziciją jam pasiūlyti komandoje, nuo kokių užduočių jis gali pradėti dirbti, kurioje komandoje darbuotojas būtų naudingesnis įmonei. Straipsnio autoriai išdiskutavo šią problemą su įmonės vadovais, kurie kaip užsakovai išdėstė reikalavimus, kuriais vadovaujantis būtų suprojektuota, o tada suprogramuota ekspertinių sistemų kūrimo ir vykdymo aplinka įmonės reikmėms.

Analizuojant modelio reikalavimus, aptariant privalomas ir papildomas funkcijas, modelio dizainą ir pačią veikimo logiką, buvo atliktas interviu su įmonės vadovu, kurio metu aptarti visi reikalavimai ir funkcijos. Interviu metu buvo išsiaiškinta, kad užsakovas nori ekspertinių sistemų kūrimo aplinkos, kurioje būtų galima sumodeliuoti bet kokią samprotavimo grandinę. Sistema neturi būti valdoma vien IT skyriaus, nes yra planų panaudoti sistemą įmonės klientų tikslų ir poreikių tyrimams, gamybos ir sandėlio valdymo sistemų tobulinimui. Sistemoje turi būti galimybė kurti, redaguoti ir vykdyti pasirinkimų medį, kuris nuvestų vartotoją prie reikalingos išvados. Sistemoje turi būti vartotojai ir jų grupės. Vartotojams arba visai vartotojų grupei priskiriamos ekspertinės sistemos, kuriose jie gali vykdyti pasirinkimus ir gauti rezultatą. Sistemos pagrindinė kalba turi būti anglų, nes įmonėje dirba darbuotojai iš kitų Europos šalių, bet sistemoje gali būti vertimai kitomis (lietuvių, vokiečių) kalbomis.

Svarbus sistemos aspektas – API ir duomenų apsikeitimas. Sistemoje turi būti galimybė nuskaityti duomenis per sistemos API, HTTP GET metodu. Tai yra svarbu, nes, jei bus dirbama su klientais šioje kuriamoje aplinkoje, kai kurie klientai gali paprašyti ekspertinės sistemos medžio XML failo, kaip teigė užsakovas. Reikalingas ir duomenų importas, kur būtų galima HTTP POST metodu įkelti ES žinių bazę pagal pateiktą XML formatą, kurio teisingumą ir struktūrą patikrintų XSD schema. Duomenų importas reikalingas, nes įmonėje programuotojai naudoja įvairias alternatyvas procesų valdymui, tad, norint subendrinti visą įmonės darbą į vieną sistemą, reikia galimybės apsikeisti duomenimis su jau naudojamomis sistemomis [13].

Ekspertinių sistemų kūrimo aplinkos

Sprendimo priėmimas yra procesas, kai pasirenkamas specifinis variantas iš galimų sąrašo norint, kad galutinis rezultatas atitiktų asmens, kuris atlieka sprendimą, norus ir tikslus [12]. Praktikoje tokie sprendimai (dar vadinami alternatyvomis) pasireiškia skirtingų kompiuterinių sistemų pasirinkimu, skirtingų žmonių specifinėmis darbo paraiškomis arba skirtingomis investavimo į atliekamą veiklą strategijomis. Sunkumai atliekant sprendimą pasitaiko daugelyje veiklos sričių, problemų šaltinis

dažniausiai slypi situacijos sudėtingume, kurį sudaro: didelis kiekis parametrų ir sąlygų, kurios lemia sprendimą; nebaigti, neaiškūs ar prieštaringi tikslai ir žinios; didelis kiekis arba neišsamiai apibrėžti variantai; skirtingos suinteresuotųjų grupės, turinčios skirtingus problemų sprendimo būdus; laiko suvaržymai, lemiantys sprendimą.

Padėti žmonėms atlikti sudėtingus sprendimus yra taikomos ekspertinių sistemų kūrimo aplinkos (angl. *Expert system shell*), kurios suteikia pagalbą ekspertams modeliuoti sudėtingus sprendimus, vystyti savo veiklos sritis ir atlikti sprendimų analitiką [6].

Ekspertinių sistemų kūrimo aplinka (toliau – ESS) yra pagrindas, naudojamas ES kūrimui. Ji neturi žinių bazės. Priklausomai nuo reikalavimų žinovas, srities specialistas arba ekspertas užpildo kūrimo aplinkos žinių bazę ESS duomenimis. Pavyzdžiui, jei žinovas užpildo žinių bazę, sukurdamas ekspertinės modelio medį apie automobilių variklių diagnostikos principus, sistema panaudos eksperto žinių banką ir elgsis lyg šios srities ekspertas, imituodama jo buvimą, kai jai bus užduodami klausimai arba pateikiami faktai, laukiant atsakymo apie variklio diagnozę. Tokiu būdu ESS suteikia būdą sukurti greitą ekspertinę sistemą. Po to galutinis vartotojas, turėdamas prieigą prie vartotojo sąsajos, gali gauti eksperto sprendimus, pasirinkti tam tikrus ES sprendimus, klausimus ar atsakymus, priklausomai nuo eksperto pateikto ES medžio. Priklausomai nuo ES sudėtingumo ir jos medžio gylio vartotojas gali gauti specifinį atsakymą ar sprendimą pagal jo pasirinkimų ir samprotavimų logiką. Ekspertinė sistema gauna vartotojo įvestį ir pagal jo pasirinkimų šabloną iš žinių bazės ištraukia atitinkamą taisyklę arba pabaigą. Žinių bazės redaktorius suteikia redagavimo galimybes, kurios leidžia greitai papildyti žinių bazę modifikuotomis arba visiškai naujomis taisyklėmis. Redagavimas vyksta pagal ES esančią samprotavimo medžio tvarką. Jei nauja taisyklė sutampa su keliais jau ES turimais šablonais, tada bus gauta ta pati pabaiga. Sistema visada turi dilemą: atmesti arba pasirinkti sprendimo šabloną. Pasirinkimo atveju naudojami algoritmai, tokie kaip „Rete", kurie nusprendžia, kokia sprendimo eiga palankiausia. Tai vadinama „*konfliktų sprendimu*". Kai sistema pasirenka tinkamiausią šabloną, galutinis rezultatas išvedamas vartotojui modelio vartotojo sąsajoje [7].

Sistemos realizavimo įrankių pasirinkimas

Interviu metu, pasitarus su įmonėje dirbančiais programuotojais ir paanalizavus mokslinius straipsnius [15, 18, 19, 29, 30, 31], buvo nuspręsta, kad geriausia kalba tokio tipo projekto realizavimui yra PHP. Todėl ir buvo pasirinkta PHP programavimo kalba. 2019 metų duomenimis 36% visų internetinių puslapių yra sukurti PHP programavimo kalba. Tai reiškia, kad ji yra viena iš palankiausiai vertinamų kalbų tarp programuotojų ir svetainių kūrėjų [15]. PHP naudojama didžiausiose ir populiariausiose bei įtakingiausiose internetinėse svetainėse ir platformose pasaulyje, tokiose kaip Facebook, Vikipedija, WordPress ir Zoom [27]. PHP yra dažnai naudojama atviro kodo scenarijų principo kalba (angl. *Scripting language*), kuri yra populiariausia internetinių aplikacijų ir puslapių kūrimui [26]. PHP yra serverio pusės programavimo kalba, įterpta į HTML savo paprasčiausioje formoje. PHP leidžia žiniatinklio kūrėjams kurti dinamišką turinį ir sąveikauti su duomenų bazėmis. PHP yra žinoma dėl savo paprastumo, greičio ir lankstumo funkcijų, dėl kurių ji tapo kertiniu internetinių sistemų kūrimo įrankiu [30]. Scenarijų kalbas, tokias kaip PHP, interpretuoja kita programa jų vykdymo metu (kompiliavimo nereikia). Scenarijų kalbos gali būti interpretuojamos serverio arba kliento pusėje – naršyklėje (angl. *Server side* ir *Client side*). PHP serverio pusėje yra scenarijų kalba, kurią žiniatinklio serveryje apdoroja PHP interpretatorius, o rezultatas (išvestis) siunčiamas į interneto naršyklę kaip paprastas HTML puslapis. PHP galima nemokamai atsisiųsti, įdiegti ir naudoti, nes tai yra atviro kodo programavimo kalba, kurią išlaiko jos bendruomenė [30]. Objektinis programavimas (OOP) naudoja „*objektus*" tam, kad juose būtų duomenys ir funkcijos, padedančios kurti sudėtingesnes, daugkartinio naudojimo internetines programas. Nagrinėjamame projekte OOP buvo pridėtas prie PHP5. PHP naudoja savo atmintį, sumažindama serverio darbo krūvį ir padidindama našumą. PHP gali būti iki 382% greitesnė nei Python ir 195% greitesnė už Ruby [15]. PHP sintaksė yra lengvai suprantama ir išmokstama, nesvarbu, ar kuriama nuo nulio, ar naudojamos esamos modelio , ar priedai. PHP palaiko visas pirmaujančias duomenų bazes (MySQL, SQLite, ODBC) ir yra suderinama su dauguma serverių (Apache, IIS ir kt.). Ji naudojama visose platformose („Windows", „Mac OS", „Linux" ir kt.), turi

daug palaikomų PHP karkasų kaip: Laravel, CodeIgniter, Symfony bei daugybę gerai aprūpintų ir patikrintų bibliotekų [15].

Kaip naujos technologijos panaudojimas ekspertinių sistemų kūrimui papildomai buvo panaudotas Laravel, PHP kalbos karkasas. Laravel pasirinktas atsižvelgiant į įmonės programuotojų rekomendaciją. Jis suteikia visas reikalingas galimybes modelio realizavimui bei galimybę įsidiegti papildomus paketus, kurie palengvina kūrimo procesą. Laravel yra labiausiai tinkama PHP sistema pradedantiesiems ir pažengusiems programuotojams. Ji gali sutrumpinti internetinių programų kūrimo ir patekimo į rinką laiką naudojant šiuolaikinius objektinius PHP metodus. Jo išraiškinga sintaksė ir modernios funkcijos yra tinkamos kūrėjams, norintiems kurti patikimas programas [16]. Karkaso naudojimas palengvina kūrimo procesą, nes jis suteikia kelis modulius su jų jungtimis. Karkasas suteikia atspirties tašką kuriant programą ir leidžia kūrėjui sutelkti dėmesį į sudėtingesnes problemas bei logikos vystymą. Laravel suteikia tokias funkcijas, kaip puikus ORM su populiariomis duomenų bazėmis ir priklausomybės injekcija, kuri yra labai lengvai keičiama ir manipuliuojama. Laravel yra pagrįstas MVC architektūra. Modelis-Peržiūra-Valdiklis (MVC, angl. *Model-View-Controller*) yra dizaino modelis, kurį sudaro trys pagrindinės dalys: modelis, atvaizdavimas ir valdiklis. Šie komponentai vienas su kitu bendrauja pagal MVC logiką: atvaizdavimo modelyje vartotojas suveda duomenis, valdiklis juos gauna ir apdoroja, nusiunčia modelio komponentui, kuris juos išsaugo. Atvaizdavimo *View* komponentas skirtas vartotojų sąsajos logikai. Valdiklio komponentas gauna įvesties duomenis ir juos apdoroja. Tai veikia kaip sąsaja tarp modelio ir atvaizdavimo komponentų. Modelio komponentas yra logika, kuri turi ryšį su duomenimis, esančiais duomenų bazėje. Tai yra pagrindinis modelio komponentas ir reiškia perduotus duomenis tarp atvaizdavimo ir valdiklio [17]. Nagrinėjamame projekte buvo atliktas Laravel palyginimas su kitais konkuruojančiais karkasais (žr. 2 lentelę).

2 lentelė

Populiariausių PHP karkasų palyginimas [17]

Karkasas	CodeIgniter	Symphony	Laravel
Išleista pirma versija	2006m.	2005m.	2011m.
Palaikoma PHP versija	>= PHP 7.4	>= PHP 8.1	>= PHP 7.4
Kodo generavimas	Nėra	CLI	CLI
ORM	ActiveRecord	Doctrine 2, Propel	ELOQUENT ORM
Šablonų sistema	PHP	PHP, Twig	PHP, Blade, pasirinktinis
Autorizacijos paketas	Nėra	Yra	Yra
SQL injection apsauga	Yra	Yra	Yra

CodeIgniter yra atviro kodo internetinių programų kūrimo karkasas, skirtas dinaminėms PHP programoms kurti. Jo pagrindinis tikslas yra padėti internetinių programų kūrėjams dirbti greičiau, nereikalaujant rašyti viso kodo nuo pradžių. CodeIgniter buvo pristatytas 2006 m. vasario 28 d. Šio karkaso charakteristikos yra šios:

- Jis pasižymi lankstumu ir lengvumu, palengvinančiu mokymąsi, bibliotekų ir pagalbinių priemonių keitimą bei integravimą;
- Jis naudoja MVC (Model-View-Controller) modelį, kuris padeda struktūrizuoti kodą ir užtikrina aiškius standartus.
- Jo URL valdymas yra labai draugiškas ir supaprastintas [19].

Siekiant kurti sudėtingas internetines programas, Sympfony sunaudoja mažiau atminties, kad užtikrintų sklandų veikimą, kai kuriamos didelio našumo internetinės programos. Sympfony yra puikus įrankis, suteikiantis daug galimybių pradedantiesiems kūrėjams. Sympfony paprasta naudoti, nes jis turi į internetinių sistemų kūrimo procesus įtrauktų dokumentuotų funkcijų, kurios didina vartotojų pasitikėjimą ir padeda greitai susigaudyti [20].

Laravel yra atvirojo kodo PHP karkasas. Kiekviena nauja Laravel versija yra patobulinta naujais „web-dev“ sprendimais ir funkcijomis. Karkasas yra žinomas savo patikimumu ir patvarumu PHP programavimo kalboje. Laravel turi labai patikimą saugos modulį ir daugelį modulių, padedančių kurti didelio našumo ir kintančio dydžio projektus. Laravel leidžia kurti daug funkcijų turinčias svetaines ir programas dėl iš anksto sukurtų paketų ir funkcijų: pvz., apsaugos funkcija ir kelių

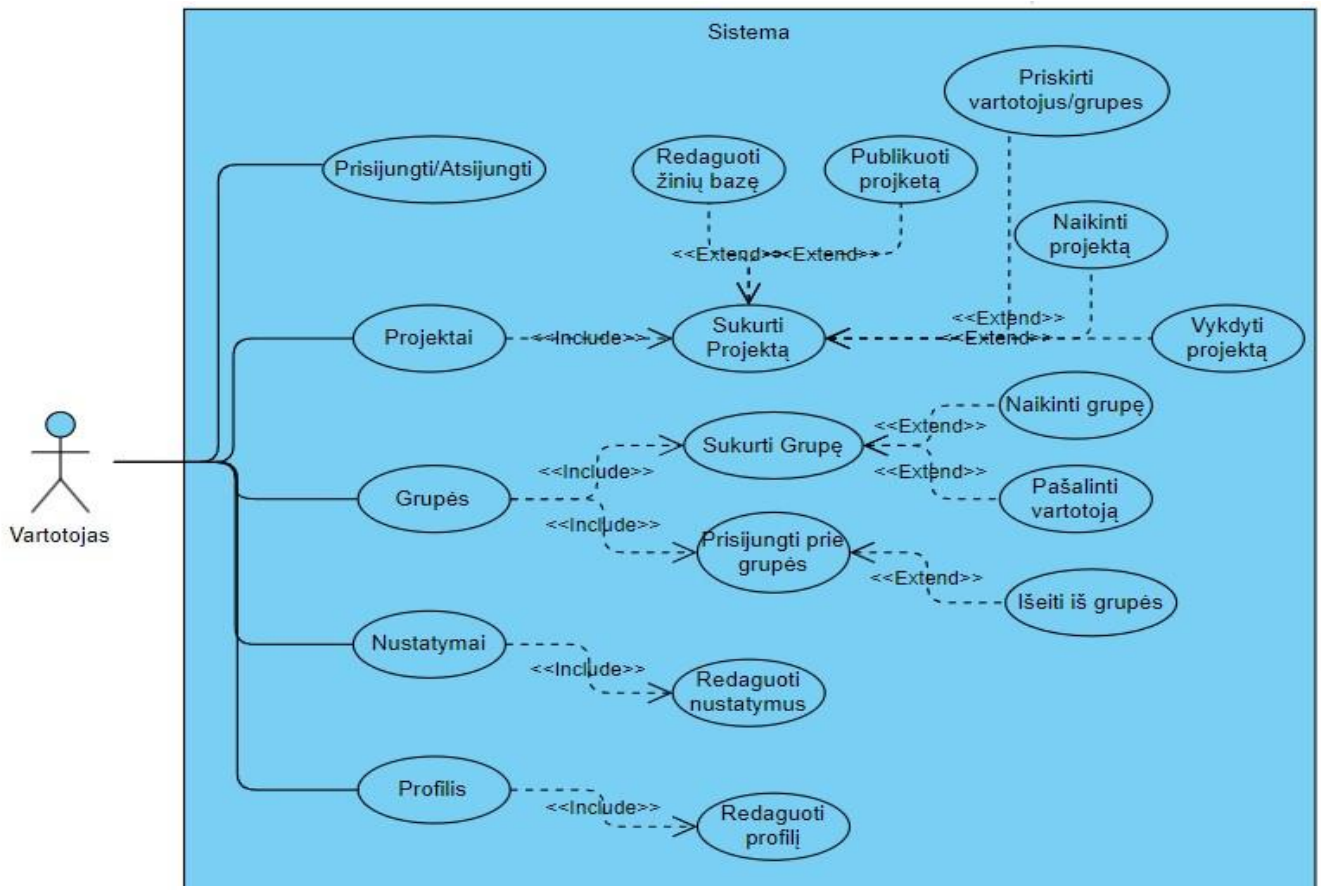
svetainių patikros apsauga, vartotojo autentifikavimo logika, supaprastinta integracija su el. pašto programomis, pranešimų eilės sistema ir laukiančios užduotys. Šių paketų ir standartinių funkcijų dėka kūrėjai neskiria daug laiko kurdami juos nuo nulio, o tiesiog diegia jau paruoštus sprendimus, sutaupydami laiko ir resursų projektų vystymui [22, 21].

Išanalizavus 3 karkasus buvo nuspręsta projekte naudoti Laravel karkasą. Visi karkasai yra panašūs vienas su kitu, bet Laravel pasižymi paprastumu ir naujausiomis technologijomis. Kadangi Laravel karkasas yra naujausias iš šio sąrašo ir buvo kurtas atsižvelgiant į Symphony karkaso trūkumus, jis puikai tinka nagrinėjamos modelio realizavimui, nes turi visas reikiamas funkcijas, pagalbą ir dokumentaciją projekto įgyvendinimui.

Programos kodo rašymui buvo naudojamas nemokamas atviro kodo redaktorius Visual studio code, kuris turi plėtinių su PHP kalba. Įdiegus šiuos plėtinius, redaktorius atpažįsta PHP kodą ir gali padėti su įvairiomis PHP funkcijomis bei tikrinti patį kodą. Serveriui realizuoti ir paleisti sistemą buvo naudojamas XAMPP paketas, kuris savyje turi PHP programavimo kalbos interpretatorių, MariaDB duomenų bazių valdymo platformą ir Apache HTTP serverio paleidimui. Šis paketas pasirinktas dėl jo patogumo: viskas centralizuota viename pakete ir nereikia papildomo administravimo įrankių. Nebereikia atskirai įdiegti PHP kalbą, duomenų bazių valdymo sistemą, viską atlieka šis paketas. Nuspręsta projekte naudoti MariaDB duomenų bazę, Apache serverį, kuris lengvai leidžia paleisti sistemą testinėje aplinkoje ir ją testuoti.

Panaudos atvejų diagrama

UML – modeliavimo ir specifikacijų kūrimo kalba, skirta specifikuoti, atvaizduoti ir konstruoti į objektus orientuotų programų dokumentus. Panaudos atvejų diagrama (Užduočių diagrama) (angl. *Use case diagram*) – UML diagrama, aprašanti, ką projektuojama sistema gali atlikti, kartu aprašydama ir išorinius sistemos veikėjus [18]. Didelėms sistemoms ši diagrama skirstoma į posistemes. Pagrindinis šios diagramos elementas – panaudos atvejis, aprašantis aibę panašių



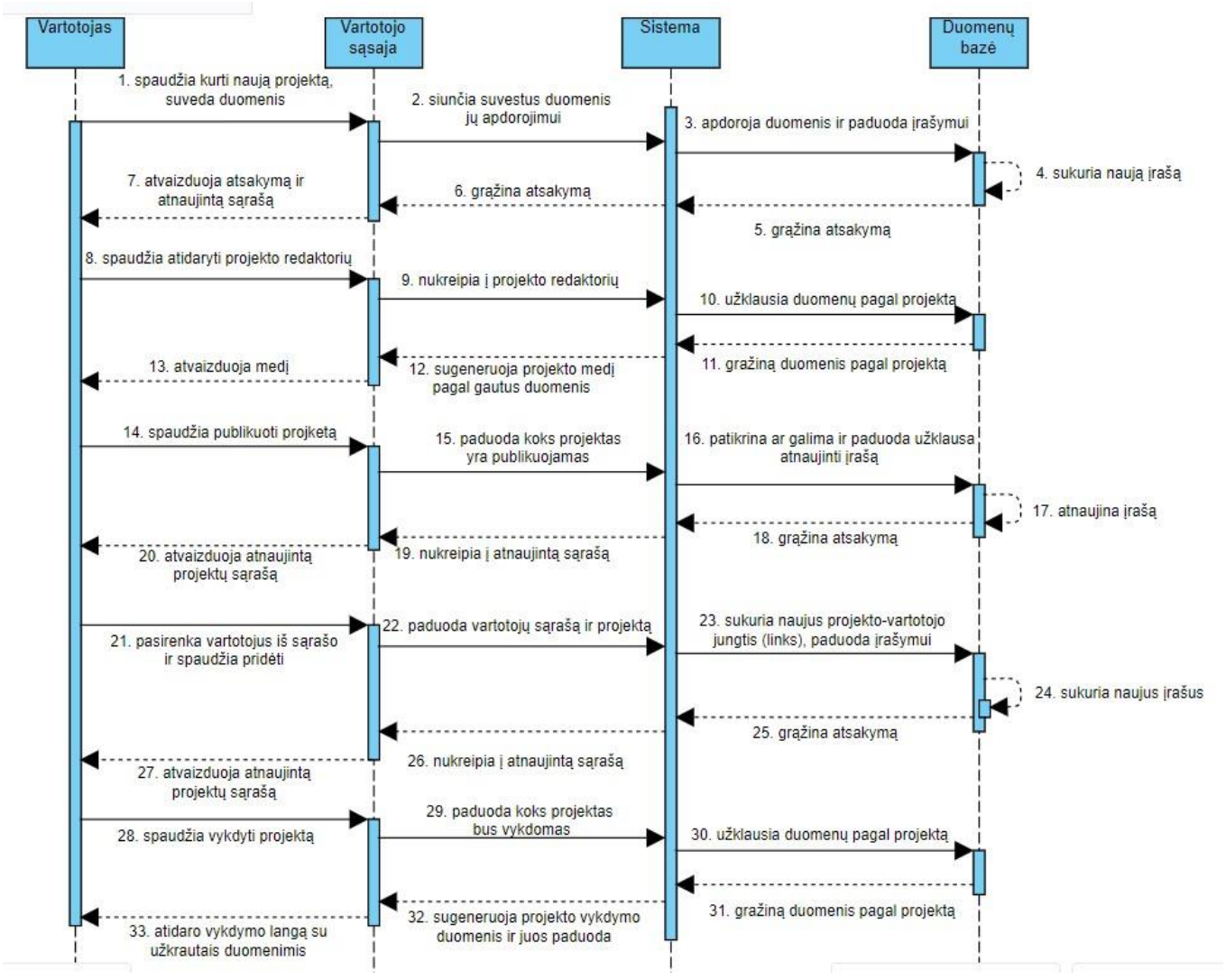
2 pav. Visos sistemos panaudos atvejų diagrama
Sudaryta autorių

sąveikos scenarijų. Kiekvienas panaudos atvejis paprastai turi vieną pagrindinį ir keletą šalutinių scenarijų, kurie aprašomi dinaminėmis UML diagramomis (sekų, bendradarbiavimo, veiklos) [18].

Panaudos atvejų diagrama (žr. 6 pav.) atvaizduoja išsamią schemą apie galimus veiksmus sistemoje. Vartotojas, prisijungęs prie sistemos, pirmiausia matys informaciją apie sistemą pagrindiniame puslapyje, kuriame bus nukreipimai į kitus sistemos puslapius. Vartotojas galės kurti projektus, grupes, redaguoti nustatymus ir profilį. Projektų sąraše vartotojas galės sukurti ekspertinės sistemos projektą, redaguoti žinių bazę, pridėti vartotojus ir kitos funkcijas. Grupių sąraše vartotojas galės kurti grupes, prisijungti prie jau sukurtų, pridėti ir pašalinti vartotojus. Nustatymuose ir profilyje vartotojas galės keisti juos tik savo vartotojo atžvilgiu, nes kiekvienas vartotojas turės savo nustatymus ir profilį.

Sekų diagrama

Sistemos funkciniai reikalavimai yra užfiksuojami panaudos atvejų diagramose. Siekdami realizuoti naudojimo atvejus, sistemos analitikai ir projektuotojai turėtų pateikti sekų diagramą. Sekos diagrama yra vienas iš UML diagramų tipų, vaizduojantis sistemoje vykstančių veiksmų seką. Sekų diagramoje fiksuojamas metodų kvietimas kiekviename objekte ir kvietimų tvarka. Sekų diagrama yra dvimatė. Horizontalioji ašis rodo vaizduojamo objekto gyvavimą, o vertikalioji ašis rodo šių objektų sukūrimo ar kvietimų seką [22, 23].

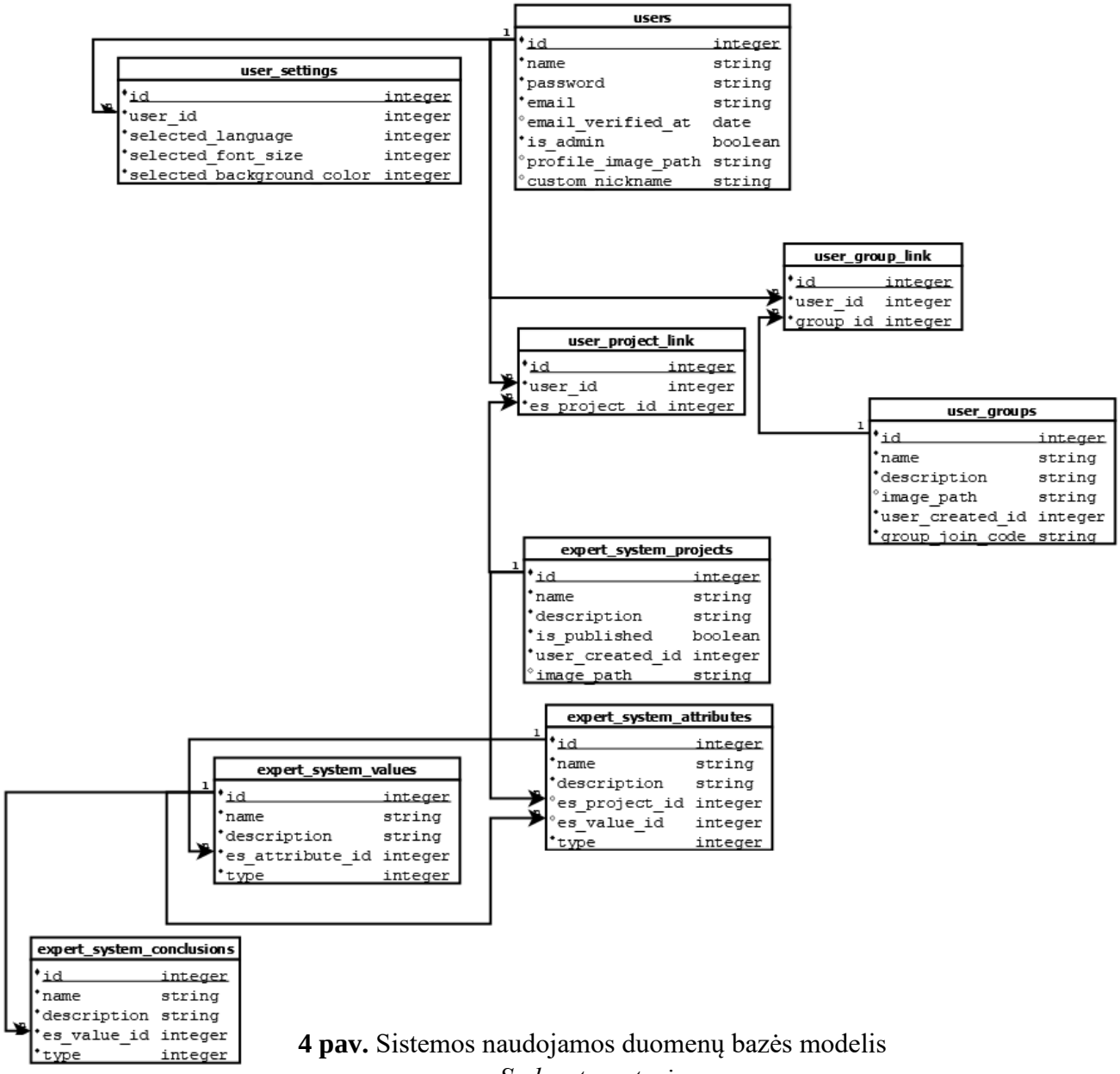


3 pav. Sistemos projekto sekų diagrama
Sudaryta autorių

7 pav. atvaizduota ekspertinių sistemų projekto sukūrimo, žinių banko užpildymo, publikavimo, vartotojų priskyrimo ir vykdymo sekų diagrama. Atvaizduotas procesas atitinka pilną ekspertinės sistemos modelio sukūrimą ir paleidimą vartotojams, kai visi duomenys tinkamai suvesti. Vartotojas susikuria projektą sistemoje, užpildydamas laukus atsidariusiame dialoge. Pirmiausia vartotojui reikia sukurti žinių bazę, samprotavimo grandinės (išsišakojančio medžio) principu. Užpildžius žinių bazę, vartotojas gali bandyti publikuoti ekspertinės sistemos projektą. Sistema patikrins medžio struktūrą, kad visos reikšmės turėtų pabaigą. Jei visos reikšmės turi pabaigą, sistema publikuos projektą. Kai projektas yra publikuotas, belieka jį priskirti vartotojams. Priskyrus projektą, vartotojai jį mato ir gali vykdyti.

Duomenų bazės modelis

Realizuoti modelio veikimui reikėjo sumodeliuoti duomenų bazę. Pagal suprojektuotą ir sutartą veikimą buvo numatyta, kad sistemai reikės lentelių saugoti vartotojų duomenis, grupių projektus ir nustatymų duomenis. Papildomai reikalingos ir pagalbinės lentelės ekspertinės sistemos medžio generavimui, duomenų bazės lentelėse turės būti saugomi ir pačios ekspertinės sistemos elementai, bei sujungimo įrašai (links), kurie sujungia projektą arba grupę su vartotojais. Sukurtą modelį galima matyti 8 pav. schemoje.



4 pav. Sistemos naudojamos duomenų bazės modelis
Sudaryta autorių

Pagrindinėse duomenų bazės lentelėse (pvz., expert_system_projects arba users) yra saugoma informacija apie pačius sisteminius objektus: pavadinimas, vardas, aprašai ir panaši informacija. Lentelės user_group_link ir user_project_link yra naudojamos sujungti vartotoją su kitu objektu, projektu arba grupe. Taip sistema galės rasti, kurioje grupėje arba kuriam projektui kokie vartotojai priskirti ir atvaizduoti sąraše. Lentelės expert_system_attributes, expert_system_values ir expert_system_conclusions naudojamos saugoti pačios ekspertinės modelio medžio elementus, kuriuos, užklausus duomenų bazės, galima naudoti redaktoriaus generavimui, atiduoti per API ir jų vykdymui.

Duomenų bazės SQL (struktūrizuotos užklausos kalba) yra labai svarbi projektuojant sistemas, kurios naudoja reliacinę duomenų saugojimo struktūrą. Štai keletas priežasčių, kodėl duomenų bazės SQL yra reikalinga sistemų projektavime:

1. *Duomenų saugojimas*: duomenų bazės SQL leidžia efektyviai ir struktūrizuotai saugoti didelius kiekius duomenų. Jos leidžia sukurti lentelės struktūrą, apibrėžti stulpelių tipus ir nustatyti ryšius tarp skirtingų lentelių. Tai suteikia galimybę tvarkingai organizuoti ir valdyti duomenis.
2. *Duomenų integravimas*: daugelis sistemų naudoja skirtingus duomenų šaltinius ir duomenų formatus. Duomenų bazės SQL leidžia integruoti duomenis iš skirtingų šaltinių, suderinant ir susiejant juos pagal bendrą raktą arba sąlygas. Tai leidžia sistemai gauti bendrą ir vienodą duomenų rinkinį, kuris gali būti naudojamas veiksmams ir analizei.
3. *Duomenų manipuliavimas*: SQL suteikia galingas užklausų funkcijas, kurios leidžia atlikti paiešką, filtravimą, rūšiavimą ir kitas operacijas su duomenų bazės įrašais. Tai leidžia lengvai ir efektyviai išgauti norimus duomenis iš duomenų bazės, atnaujinti, įterpti arba ištrinti įrašus, atlikti skaičiavimus ir daug daugiau kitų veiksmų.
4. *Duomenų vientisumas ir saugumas*: duomenų bazės SQL palaiko vientisumo taisykles ir suteikia galimybes apsaugoti duomenis nuo nepageidaujamo pakeitimo ar praradimo. Jos gali taikyti apribojimus ir taisykles duomenų įvedimui, atnaujinimui ir trynimui, užtikrinant, kad duomenys būtų tikslūs ir patikimi [25].

REST API

REST API (Representational State Transfer Application Programming Interface) yra architektūrinis stilius, naudojamas informacijos perdavimui ir bendravimui tarp skirtingų sistemų per internetą [14]. Tai yra būdas, kaip duomenys ir veiksmų užklausos yra siunčiamos ir gautos tarp kliento ir serverio. REST API naudoja HTTP protokolą ir HTTP metodus, tokius kaip GET, POST, PUT ir DELETE, norint vykdyti veiksmus su išteklių (resursų) duomenimis. Šis API yra „*stateless*“ (be būsenos), tai reiškia, kad kiekviena užklausa yra nepriklausoma ir nesaugo jokios informacijos apie ankstesnes užklausas [24]. Išnagrinėjus technologinius sprendimus ekspertinių sistemų kūrimui, buvo pastebėta kuriant ES nėra taikomas REST API architektūrinis stilius. Dažniausiai jis naudojamas esamiems duomenims iš interneto šaltinių paėmimui [29,30,31]. Sistemos modelyje realizuotas ir API modelis, kuris suteikia galimybę apsikeisti duomenimis tarp dviejų sistemų. Kuriant modelio API, buvo atsižvelgta į saugumą dėl duomenų pasisavinimo ir apsaugą nuo nereikalingo duomenų siuntinėjimo. To realizavimui modelio kode buvo nurodyti tik tie klientų IP adresai, iš kurių bus galima pasiekti sistemą ir jos API. REST API modulyje realizuotos duomenų eksporto ir importo funkcijos. Apsikeitimas duomenimis, kaip buvo išanalizuota poreikių reikalavime, vyks XML failais. Užklausiant duomenų pagal projekto pavadinimą, sistema pirmiausia pasitikrins, ar yra toks projektas. Jei projektas yra, sistema sugeneruoja XML failą, naudodama rekursines funkcijas sugeneruoti modelio medį, tada tą medį paverčia į XML failą ir tą failą atiduoda HTTP užklausos „response body“ dalyje. Vartotojas gali pasiimti ir naudoti per užklausą gautus duomenis. Sukurtas sistemos modelis gali panaudoti kelių sukurtų projektų duomenis, jei tarp jų yra ryšys. Sistemos modelis gali nuspręsti (naudojant tam tikrus svorio koeficientus), ar panaudoti kitos sistemos duomenis. REST API naudojamas siųsti duomenis iš ekspertinės sistemos į kitą ekspertinę sistemą ir grąžinti rezultatus vartotojui. Svarbu užtikrinti, kad visi komponentai būtų tinkamai sukonfigūruoti ir turėtų suderintą sąsają.

Duomenų importavimo funkcija modelio projektui buvo įgyvendinta, leidžiant vartotojui perduoti XML failą naudojant HTTP POST metodą. Paduodamas XML failas turi būti konkrečios struktūros ir atitikti modelio reikalavimus. XML failas turi atitikti XSD validacijos struktūrą ir po to būti patikrintas modelio logikos validacijos atžvilgiu (pvz., patikrinti projektų pavadinimo unikalumą). Priklausomai nuo paduoto failo, sistema gali grąžinti klaidos arba sėkmės kodą. Jei duomenys buvo teisingi ir sistema sėkmingai nuskaitė failą bei sukūrė naują projektą, vartotojui, kuris siuntė užklausą, bus grąžintas pranešimas apie sėkmingą importavimą.

Nagrinėjant technologinius sprendimus ekspertinių sistemų kūrimui, pastebėta, kad REST API architektūrinis stilius nenaudojamas ekspertinių sistemų kūrimui, jis naudojamas tik duomenų panaudojimui iš kitų šaltinių. Šiame tyrime sukurtas sistemos modelis, įgyvendinantis REST API ekspertinėje sistemoje. Modelyje atsižvelgta ir į saugumo aspektus, nustatant leidimus tik tam tikriems klientų IP adresams.

Išvados

Atsižvelgiant į iškeltus įmonės modelio kūrimo reikalavimus, atlikta ekspertinių sistemų ir jos kūrimo aplinkų analizė. Išanalizuoti moksliniai straipsniai, knygos, remiantis jose pateiktomis ir siūlomomis idėjomis, sprendimais ir metodais buvo parinkti technologiniai sprendimai modelio projektui. Darbo metu buvo atlikta ekspertinių sistemų ir jos kūrimo aplinkų analizė.

Realizuojant ekspertinių sistemų kūrimo įrankio/platformos modelio įgyvendinimą, buvo atliktas interviu protokolas su užsakovu. Interviu metu su klientu sutarti modelio funkciniai reikalavimai ir dizainas. Įmonės poreikiai yra susiję su naujų darbuotojų mokymu, sprendimų priėmimu ir duomenų valdymu. Modelio reikalavimai apima galimybę modeliuoti bet kokią samprotavimo grandinę, valdyti vartotojų ir grupių sąrašus bei įgyvendinti duomenų importą ir eksportą. Išanalizuoti modelio kūrimo įrankiai ir, pasitelkus profesionalių įmonės programuotojų patarimus, pasirinkti kūrimo įrankiai/platformos įgyvendinimui.

Sistemos modelyje įgyvendintos duomenų eksporto ir importo funkcijos, kurios leidžia apsikeisti duomenimis tarp sistemų naudojant XML failus. Sukurtas modelis gali panaudoti kelių projektų duomenis, priklausomai nuo ryšių ir svorio koeficientų. REST API yra naudojamas siųsti duomenis iš ekspertinės sistemos į kitą sistemą ir grąžinti rezultatus vartotojui. Analizė ir išvados leidžia teigti, kad numatytas ekspertinių sistemų kūrimo projektas turi aiškius tikslus, apibrėžtus poreikius ir tinkamai pasirinktus technologinius įrankius, kuriuos galima sėkmingai įgyvendinti, atsižvelgiant į įmonės specifiką ir poreikius.

Literatūra

1. BOHANEC, Marko; RAJKOVIC, V. Expert system for decision making. *Sistemica*, 1990, 1(1): 145-157. http://www-ai.ijs.si/MarkoBohanec/pub/IFIP1983.pdf
2. JANULEVIČIUS, Justinas; GORANIN, Nikolai. Expert system for data security risk management for SMEs. *Mokslas – Lietuvos Ateitis / Science – Future of Lithuania*, 2013, 5(2): 84-87. DOI: https://doi.org/10.3846/mla.2013.14
3. GRABOWSKI, Barbara; HARKNESS, William. Enhancing Statistics Education with Expert Systems: More than an Advisory System. *Journal of Statistics Education*, 2017, 4(3). DOI: 10.1080/10691898.1996.11910514
4. PEREIRA, Clara; SILVA, João N.; SILVA, Ana; DE BRITO, Jorge; SILVESTRE, Jose D. Building Inspection System Software Based on Expert Knowledge. *Journal of Performance of Constructed Facilities*. 2022, 36(2): 1-13. DOI: 10.1061/(ASCE)CF.1943-5509.0001700
5. Tan, C. F., Wahidin, L. S., Khalil, S. N., Tamaldin, N., Hu, J., & Rauterberg, G.W. M. (2016). The application of expert system: A review of research and applications. ARPN
6. Journal of Engineering and Applied Sciences, 11(4), 2448-2453.)
7. BOHANEC, Marko; RAJKOVIČ, Vladislav. DEX: An expert system shell for decision support. *Sistemica*, 1990, 1(1): 145-157. https://kt.ijs.si/MarkoBohanec/pub/Sistemica90.pdf

8. KUMAR, Sanjay; PRASAD, Rajkishore. Importance of expert system shell in development of expert system. *Intern. Journal of Innovative Research & Development*, 2015, 4(3): 128-133. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=abd708b81c41a0c61028813565bfae893d722a0f
9. HORVITZ, Eric J.; BREESE, John S.; HENRION, Max. Decision theory in expert systems and artificial intelligence. *International journal of approximate reasoning*, 1988, 2(3): 247-302. https://www.sciencedirect.com/science/article/pii/0888613X8890120X
10. ARIAS-ARANDA, Daniel, et al. A fuzzy expert system for business management. *Expert Systems with Applications*, 2010, 37(12): 7570-7580. https://doi.org/10.1016/j.eswa.2010.04.086
11. IVANOV, Andrey; KORABLEVA, Galina. The Results of Development and Appliance of an Expert System for Public Catering Businesses' Competitive Index Assessment. In *XV International Scientific Conference "INTERAGROMASH 2022" Global Precision Ag Innovation 2022*, Vol. 1, 1267-1281. Cham: Springer International Publishing, 2023. https://link.springer.com/chapter/10.1007/978-3-031-21432-5_134
12. WATERMAN, Donald A. *A guide to expert systems*. Addison-Wesley Longman Publishing Co., Inc., 1985.
13. EFSTATHIOU, J.; MAMDANI, E. H. Expert systems and how they are applied to industrial decision making. *Computer assisted decision making*. Amsterdam: Elsevier, 1986.
14. GU, Xiaodong, et al. Deep API learning. In: *Proceedings of the 2016 24th ACM SIGSOFT international symposium on foundations of software engineering*. 2016, 631-642. https://arxiv.org/pdf/1605.08535.pdf
15. MUMBAIKAR, Snehal, et al. Web Services Based on SOAP and REST Principles. *International Journal of Scientific and Research Publications*, 2013, 3(5): 1-4.
16. LAAZIRI, Majida, et al. A comparative study of laravel and symfony PHP frameworks. *International Journal of Electrical and Computer Engineering*, 2019, 9(1): 704-712.
17. PECORARO, Christopher John. *Mastering Laravel*. Packt Publishing, 2015.
18. SUBECZ, Zoltán. Web-development with Laravel framework. *Gradus*, 2021, 8(1): 211-218. http://real.mtak.hu/125616/1/2021_1_CSC_006_Subecz.pdf
19. DAUGIRDAS, Donatas; PRANCIULIS, Laisvūnas; ČEPIS, Aidas. Projection and implementation of a model for a game rental system. *Applied Scientific Research*, 2023, 2(1): 65-73. https://doi.org/10.56131/tmt.2023.2.1.99
20. HIMAWAN, Albert Kurnia. Performance Analysis Framework Codeigniter and CakePHP in Website Creation. *International Journal of Computer Applications*, 2014, 94(20), 6-11. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8b56e38927d1097f429ae6330f51e08ff3e9cb8e
21. LANCOR, Lisa; KATHA, Samyukta. Analyzing PHP frameworks for use in a project-based software engineering course. In: *Proceeding of the 44th ACM technical symposium on Computer science education*, 2013, 519-524. https://doi.org/10.1145/2445196.2445350
22. HE, Ren Yu. Design and implementation of web based on Laravel framework. In: *2014 International Conference on Computer Science and Electronic Technology (ICCSET 2014)*. Atlantis Press, 2015, 301-304. https://www.atlantis-press.com/proceedings/iccset-14/16334
23. BELL, Donald. UML basics: The sequence diagram. 2004, 1-24. https://www.csun.edu/~twang/380/Slides/SequenceDiagram.pdf
24. THUNG, Ferdian, et al. Network Structure of Social Coding in GitHub. In: *2013 17th European conference on software maintenance and reengineering*. IEEE, 2013, 323-326. DOI: 10.1109/CSMR.2013.41
25. RIORDAN, Rebecca M. *Designing Effective Database Systems*. Addison-Wesley Professional, 2005.
26. PRETTYMAN, Steve. *Learn PHP 8*. Apress, 2020.

27. PURER, Klaus. *PHP vs. Python vs. Ruby–The web scripting language shootout*. Vienna University of Technology, 2009.
28. SAHIN, Seda; TOLUN, Mehmet R.; HASSANPOUR, Reza. Hybrid expert systems: A survey of current approaches and applications. *Expert Systems with Applications*, 2012, 39.4: 4609-4617. https://www.sciencedirect.com/science/article/abs/pii/S0957417411012619
29. WENG, Bin, et al. Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications*, 2018, 112: 258-273. https://www.sciencedirect.com/science/article/abs/pii/S0957417418303622
30. AYVAZ, Serkan; ALPAY, Koray. Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, 2021, 173: 114598. https://www.sciencedirect.com/science/article/abs/pii/S0957417421000397
31. WALEK, Bogdan; FOJTIK, Vladimir. A hybrid recommender system for recommending relevant movies using an expert system. *Expert Systems with Applications*, 2020, 158: 113452. https://www.sciencedirect.com/science/article/abs/pii/S0957417420302761