



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Ovidijus Kavaliauskas

NAUDOTŲ AUTOMOBILIO DALIŲ SANDĖLIO VALDYMO
SISTEMA

Baigiamasis magistro projektas

Vadovas

Prof. dr. Vacius Jusas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

**NAUDOTŲ AUTOMOBILIO DALIŲ SANDĖLIO VALDYMO
SISTEMA**

Baigiamasis magistro projektas
Programų sistemų inžinerija (kodas 621E16001)

Vadovas

Prof. dr. Vacius Jusas

Recenzentas

Prof. dr. Rimantas Butleris

Projektą atliko

Ovidijus Kavaliauskas

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Ovidijus Kavaliauskas

(Studento vardas, pavardė)

Programų sistemų inžinerija (kodas 621E16001)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Naudotų automobilio dalių sandėlio valdymo sistema“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. gegužės 21 d.

Kaunas

Patvirtinu, kad mano, **Ovidijaus Kavaliausko**, baigiamasis projektas tema „Naudotų automobilio dalių sandėlio valdymo sistema“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

TURINYS

| | |
|--|----|
| SANTRAUKA | 7 |
| SUMMARY | 8 |
| 1. ĮVADAS..... | 9 |
| 2. ANALITINĖ DALIS..... | 10 |
| 2.1. Egzistuojantys sprendimai | 10 |
| 2.1.1. Equinox Europe Vision | 10 |
| 2.1.2. ACCELLOS sandėlio valdymo sistema | 10 |
| 2.2. PREKIŲ ŽYMĖJIMAS SANDĖLYJE | 12 |
| 2.2.1. Prekių (detalių) žymėjimas..... | 12 |
| 2.2.2. Vartotojo sąsajos problemos..... | 14 |
| 2.3. MVC Architektūros Modelis | 17 |
| 2.4. Išvados | 20 |
| 3. PROJEKTINĖ DALIS | 21 |
| 3.1. Projekto įgyvendinimo planai | 21 |
| 3.1.1. Darbų grafikas | 21 |
| 3.1.2. Projektavimo metodika..... | 21 |
| 3.1.3. Programinė įranga..... | 21 |
| 3.1.4. Biuro įranga | 22 |
| 3.1.5. Projekto vykdytojai | 22 |
| 3.1.6. Biudžetas | 23 |
| 3.1.7. Kokybės vertinimas | 23 |
| 3.2. Sistemos apibūdinimas | 23 |
| 3.2.1. Programų sistemos funkcijos..... | 23 |
| 3.2.2. Sistemos kontekstas..... | 24 |
| 3.2.3. Vartotojo charakteristikos | 24 |
| 3.2.4. Vartotojo problemos | 24 |
| 3.2.5. Vartotojo tikslai | 24 |
| 3.2.6. Bendri apribojimai..... | 24 |
| 3.3. Sistemos paskirtis | 25 |
| 3.4. Sistemos tikslai | 25 |
| 3.5. Išvados | 25 |
| 4. REIKALAVIMŲ SPECIFIKACIJA | 26 |
| 4.1. Užsakovas ir sistemos vartotojai..... | 26 |
| 4.1.1. Užsakovas..... | 26 |
| 4.1.2. Vartotojai | 26 |
| 4.2. Įpareigojantys apribojimai | 27 |
| 4.2.1. Apribojimai sprendimui..... | 27 |
| 4.2.2. Diegimo aplinka | 27 |
| 4.2.3. Bendradarbiaujančios sistemos | 27 |
| 4.2.4. Komerciniai specializuoti programų paketai..... | 27 |
| 4.2.5. Numatoma darbo vietos aplinka..... | 27 |
| 4.2.6. Sistemos kūrimo terminai..... | 27 |
| 4.2.7. Sistemos kūrimo biudžetas | 27 |
| 4.3. Svarbūs faktai ir prielaidos | 28 |
| 4.3.1. Svarbūs faktai | 28 |
| 4.3.2. Prielaidos | 28 |
| 4.4. Veiklos sudėtis..... | 29 |
| 4.4.1. Veiklos kontekstas..... | 29 |
| 4.4.2. Veiklos padalijimas | 29 |
| 4.5. Sistemos sudėtis..... | 31 |

| | | |
|---------|--|----|
| 4.5.1. | Sistemos ribos..... | 31 |
| 4.5.2. | Panaudojimo atvejų sąrašas | 32 |
| 4.6. | Funkciniai reikalavimai ir reikalavimai duomenims | 37 |
| 4.6.1. | Funkciniai reikalavimai | 37 |
| 4.6.2. | Reikalavimai duomenims | 45 |
| 4.7. | Reikalavimai sistemos išvaizdai | 46 |
| 4.8. | Reikalavimai panaudojamumui | 46 |
| 4.9. | Reikalavimai vykdymo charakteristikoms..... | 47 |
| 4.10. | Reikalavimai veikimo sąlygoms | 47 |
| 4.11. | Reikalavimai sistemos priežiūrai | 48 |
| 4.12. | Reikalavimai saugumui | 49 |
| 4.13. | Kultūriniai-politiniai reikalavimai..... | 50 |
| 4.14. | Teisiniai reikalavimai | 50 |
| 4.15. | Naujos problemos..... | 51 |
| 4.15.1. | Problemos diegimo aplinkai | 51 |
| 4.15.2. | Įtaka jau instaliuotoms sistemoms | 51 |
| 4.15.3. | Neigiamas vartotojų nusiteikimas..... | 51 |
| 4.15.4. | Kliudantys diegimo aplinkos apribojimai..... | 51 |
| 4.15.5. | Galimos naujos sistemos sukeltos problemos..... | 51 |
| 4.16. | Uždaviniai | 51 |
| 4.16.1. | Sistemos pateikimo žingsniai (etapai) | 51 |
| 4.16.2. | Vystymo etapai | 52 |
| 4.17. | Pritaikymas..... | 52 |
| 4.17.1. | Reikalavimai esamų duomenų perkėlimui..... | 52 |
| 4.17.2. | Reikalingas duomenų transformavimas perkeliant į naują sistemą | 52 |
| 4.18. | Rizikos..... | 52 |
| 4.18.1. | Galimos sistemos kūrimo rizikos..... | 52 |
| 4.18.2. | Atsitiktinumų (rizikų) planas | 53 |
| 4.19. | Vartotojo dokumentacija ir apmokymas | 53 |
| 4.20. | Perspektyviniai reikalavimai | 53 |
| 4.21. | Išvados..... | 53 |
| 5. | ARCHITEKTŪROS SPECIFIKACIJA | 54 |
| 5.1. | Architektūros pateikimas | 54 |
| 5.2. | Architektūros tikslai ir apribojimai..... | 54 |
| 5.3. | Panaudojimo atvejų vaizdas..... | 55 |
| 5.4. | Sistemos statinis vaizdas | 55 |
| 5.4.1. | Apžvalga..... | 55 |
| 5.4.2. | Paketų detalizavimas | 56 |
| 5.5. | Sistemos dinaminis vaizdas | 59 |
| 5.6. | Išdėstymo vaizdas | 64 |
| 5.7. | Duomenų vaizdas..... | 65 |
| 5.8. | Išvados | 65 |
| 6. | TESTAVIMO MEDŽIAGA | 66 |
| 6.1. | Testavimo planas | 66 |
| 6.1.1. | Testuojama programinė įranga | 66 |
| 6.1.2. | Vienetų testavimas..... | 68 |
| 6.1.3. | Priėmimo testavimas | 68 |
| 6.1.4. | Aukšto lygio testavimas | 68 |
| 6.2. | Testavimo resursai | 68 |
| 6.3. | Testavimo įrankiai ir aplinka | 68 |
| 6.4. | Testavimo procedūros..... | 68 |
| 6.4.1. | Priėmimo testavimas | 72 |

| | | |
|--------|--|-----|
| 6.4.2. | Aukšto lygio testavimas | 72 |
| 6.4.3. | Testavimo resursų paskirstymas | 72 |
| 6.5. | Testavimo rezultatai | 72 |
| 6.5.1. | Rankinio testavimo duomenys ir rezultatai | 72 |
| 6.6. | Išvados | 76 |
| 7. | ŽODŽIŲ PAIEŠKA SQL PAGRINDO RELIACINĖSE DUOMENŲ BAZĖSE (TYRIMAS) 77 | 77 |
| 7.1. | Įvadas | 77 |
| 7.2. | Žodžių paieška duomenų bazėje | 79 |
| 7.3. | Žodžio paieškos algoritmų reliacinėmis duomenų bazėmis tyrimas | 83 |
| 7.3.1. | Duomenų bazės projektavimo Nr. 1 tyrimas | 83 |
| 7.3.2. | Duomenų bazės projektavimo Nr. 2 tyrimas | 84 |
| 7.3.3. | Duomenų bazės projektavimo Nr. 3 tyrimas | 84 |
| 7.3.4. | Duomenų bazės projektavimo Nr. 4 tyrimas | 85 |
| 7.3.5. | Duomenų bazės projektavimo Nr. 5 tyrimas | 86 |
| 7.4. | Tyrimo santrauka | 86 |
| 7.5. | Tyrimo išvados | 88 |
| 8. | KOKYBĖS VERTINIMO ATASKAITA..... | 89 |
| 8.1. | Įvadas | 89 |
| 8.2. | Realiai atlikto darbo kokybės analizės tikslai..... | 89 |
| 8.2.1. | Aptikti klaidas funkcionavime, logikoje, realizacijoje..... | 89 |
| 8.2.2. | Patikrinti ar sistema atitinka reikalavimų specifikaciją..... | 89 |
| 8.3. | Kokybės vertinimo procesas..... | 89 |
| 8.3.1. | Peržiūros | 89 |
| 8.3.2. | Interviu su užsakovu..... | 89 |
| 8.3.3. | Projektavimo komandos narių peržiūrų aprašymas..... | 89 |
| 8.3.4. | Vaidmenys ir atsakomybės..... | 89 |
| 8.3.5. | Programinio kodo peržiūra | 90 |
| 8.3.6. | Vertinimo rezultatai | 90 |
| 8.3.7. | Darbų našumas | 90 |
| 8.4. | Išvados | 90 |
| 9. | IŠVADOS..... | 91 |
| 10. | LITERATŪRA | 92 |
| 11. | PRIEDAI | 94 |
| 11.1. | 1 Priedas – Vartotojo dokumentacija | 94 |
| 11.2. | 2 Priedas – Terminų ir santrumpų žodynas | 109 |

Kavaliauskas, Ovidijus. Naudotų automobilio dalių sandėlio valdymo sistema. Magistro baigiamasis projektas / vadovas prof. dr. Vacius Jusas; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Technologiniai mokslai, programų inžinerija

Reikšminiai žodžiai: *naudotų automobilio dalių sandėlio valdymo sistema, sandėlio valdymo sistema, MVC architektūra, Laravel5, MySql duomenų bazė, reliacinė duomenų bazė, žodžio paieška reliacinėje duomenų bazėje.*

Kaunas, 2017. 109 p.

SANTRAUKA

Šiuo metu yra sukurta daugybė universalių sandėlio valdymo sistemų, tačiau įvertinus trikdžius su kuriais susiduria naudotomis automobilių dalimis prekiaujančios buvo nuspręsta sukurti specializuotą sandėlio valdymo sistemą skirtą būtent naudotų automobilio dalių prekyba užsiimančioms įmonėms.

Šiame dokumente aprašytas darbas susideda iš keturių pagrindinių dalių, pirmoje dalyje analizuojamos egzistuojančios sandėlio valdymo sistemos, apžvelgiamos sistemos įgyvendinimo problemos ir pasirenkama sistemos architektūra. Antrojoje (projektinėje) dokumento dalyje aprašoma kuriama sistema jos projektavimo metodologija, bei numatomi tikslai. Trečiame skyriuje pateikiami sistemos funkciniai ir nefunkciniai reikalavimai. Ketvirtame skyriuje pateikiama detali architektūros specifikacija, statinis sistemos vaizdas.

Septintame darbo skyriuje detalizuojami atlikti tyrimai ir eksperimentai, tyrimai buvo vykdomi atliekant žodžių paiešką SQL pagrindo reliacinėse duomenų bazėse.

Kavaliauskas, Ovidijus. Used Car Parts Warehouse Management System: Master's thesis in Software Engineering. Study Programme (621E16001) / supervisor assoc. prof. Vacius Jusas. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Technological sciences, software engineering

Key words: *Used car parts warehouse management system, warehouse management system, MVC Architecture, Laravel5, MySQL database, relational database, Word Search in relational database.*

Kaunas, 2017. 109 pages.

SUMMARY

Currently, there are designed numerous universal warehouse management systems, however, having assessed the various interference which are faced by the ones selling used car parts, it was decided to create a specialised warehouse management system specifically for companies engaged in trade of used car parts.

The work described in this paper consists of four main parts. The first part analyses the existing warehouse management systems, provides an overview of the existing implementation issues and selects system architecture. The second (project) part of the paper describes the developed system and its design methodology as well as foresees objectives. The third section provided functional and non-functional requirements. The fourth section contains detailed architecture specification and static image of the system.

The seventh section of the paper describes carried out research and experiments. The research were carried out using keyword search in SQL – based relational databases.

1. ĮVADAS

Yra žinoma, kad viena aktualiausių įmonės valdymo sričių yra sandėlių valdymas. Net 59 – 90 proc. įmonių mano, kad sandėliavimas yra svarbi jų verslo dalis beveik pusei jų svarbu šiuo metu didinti šios veiklos efektyvumą, tačiau tik 13 proc. įmonių naudoja tam pritaikytas programas. Net jei sandėlio valdymas nekelia didelių trukdžių patobulinus jo veiklą, ne tik padidėtų sandėlio veiklos efektyvumas, bet ir sumažėtų sandėlyje atliekamų operacijų laikas. Įmonės, įsidiegusios sandėlio valdymo sistemą, ima matyti daugiau informacijos, suteikiančios galimybę giliau, įvairesniais parametrais analizuoti logistinės grandinės procesus. Ši informacija paskatina įmones ir toliau kompiuterizuoti savo veiklas.

Naudotomis automobilių dalimis prekiaujančios įmonės susiduria su tomis pačiomis problemomis, kai klientas paskambina pasiteirauti ieškomos detalės ir pardavėjas iš karto negali pasakyti ar turi pageidaujamą prekę kur ji yra ir kiek kainuoja, todėl įmonės praranda užsakymus, mažėja apyvarta, detalės ilgiau užsilaiko sandėlyje.

Įvertinus trikdžius atsirandančius sandėlio valdymo metu buvo nuspręsta sukurti specializuotą sandėlio valdymo sistemą skirtą būtent naudotomis automobilių dalimis, kuri tuo užsiimančiai įmonei padėtų sklandžiau vykdyti prekių judėjimą, visada turėti tvarkingus ir aiškius sandėlius, greičiau patenkinti kliento poreikius, greičiau ir tiksliau vykdyti užsakymus, aptarnauti daugiau klientų su ta pačia darbo jėga.

Kalbant apie sandėlio valdymą – tai yra atsargų ir informacijos srautų valdymas, kuris apima: prekių priėmimą, užsakymų paruošimą, prekių patalpimą, kiekio ir vertės apskaitą, turimų patalpų panaudojimą, prekių išdavimą, atsargų kontrolę, informacijos perdavimą bei našų darbuotojų panaudojimą. Įmonėje nenaudojant sandėlio valdymo dažnai kliento užsakymas neįvykdomas dėl pasiklydusių prekių, kurių faktiškai nerandama atkrovimo metu arba yra sumaišomos, o vėliau brangiai kainuojantys darbuotojai ieško šių prekių.

2. ANALITINĖ DALIS

2.1. EGZISTUOJANTYS SPRENDIMAI

Nors yra sukurta daugybė sandėlio valdymo sistemų, kurios efektyviai leidžia valdyti sandėlio procesus, tačiau norėčiau išskirti dvi pagrindines sistemas: „Vision“ ir „Accellos“

2.1.1. Equinox Europe Vision

„Equinox Europe“ tiekia produktus ir paslaugas tiek korporacinėms, tiek smulkioms bendrovėms visame pasaulyje: didmenininkams, mažmenininkams, paskirstymo centrums, transporto kompanijoms. Įmonė klientams siūlo platų paslaugų spektrą, užtikrindama didelį sprendimų lankstumą bei aukštą jų kokybę.[1]

Vision – tai universalus tiekimo grandies valdymo sprendimas didmeniniams paskirstymo centrums, taip pat ir sandėliams bei tarpinėms logistikos įmonėms. Būdamą lengvai suderinama su kitais „Equinox“ produktais, sistema puikiai tinka krovinių pervežėjams ir konsolidatoriams. „Vision“ architektūra leidžia įrangą pateikti kaip atskirą universalų produktą, arba priderinti kiekvienam klientui pagal jo specifinius poreikius.[1]

2.1.2. ACCELLOS sandėlio valdymo sistema

„Accellos“ sandėlio valdymo sistemos privalumai:

Web sąsaja pagrįstos technologijos

Sandėlio dispečerio darbo sąsaja, sukurta saityno principu, užtikrina patikimą ir greitą prisijungimą prie sistemos iš bet kurio kompiuterio, prijungto prie Interneto.[2]

Lengva integracija su įmonės finansinės apskaitos sistema ir kitais išoriniais moduliais

„ACCELLOS WMS“ yra svarbiausias įmonės sandėlio valdymo sistemos elementas. „ACCELLOS WMS“ gali būti lengvai susietas su įmonės finansinės apskaitos sistema ir tuo pačiu užtikrinti sklandų darbą tarp sandėlio ir pardavimo užsakymų priėmimo modulių, atsargų valdymo ir pirkimo užsakymų. Yra padaryta ne viena integracija su „Microsoft Business Solutions: Navision, Axapta“.[2]

Mobilių delninių kompiuterinių terminalų palaikymas

„ACCELLOS WMS“ pritaikytas darbui su garsiausių pasaulio gamintojų, tokių kaip „Symbol“, „Psion Teklogix“, Unitech mobiliais terminalais. [2]

Brūkšninių kodų EAN/UPC technologijos palaikymas

Automatinio atpažinimo technologija, besiremianti produkcijos ir sandėliavimo vietų brūkšniniu kodavimu, yra pagrindinė „ACCELLOS WMS“ ideologija ir užtikrina tokius sistemos pranašumus: greitaveiką, tikslumą, patikimumą. Informacijos surinkimui ir operatyviam jos perdavimui į sistemą, „ACCELLOS WMS“ naudoja mobilius radijo terminalus su integruotais brūkšninių kodų skaitytuvais[2]

Sandėlio zonų paskirstymas

„ACCELLOS WMS“ įgalina sandėlyje sukurti iki 26 zonų su skirtingomis prekių taisyklėmis. Zonos gali atspindėti loginę sandėlio darbo organizaciją (priėmimo zona, atrinkimo zona su skirtingo atrinkimo būdais ir pan.). Taip pat galima sukurti zonas skirtingiems prekių tipams (zona prekėms padėkluose, zona prekėms dėžėse, zona vienetinėms prekėms ir pan.). Be to, galima kurti zonas prekių su atitinkamais požymiais saugojimui (degių prekių zona, šaldytų prekių zona, sunkių padėklų zona ir pan.).[2]

Prekių su skirtingais dydžio matais valdymas

„ACCELLOS WMS“ palaiko įvairius vienos prekės dydžio matus (padėklas, pakuotė, maža pakuotė, vienetas ir pan.), tuo palengvindamas prekių atrinkimo operaciją (didelė pakuotė gali būti išskaidyta, jeigu to reikia). Sandėlyje gali būti sukurtos zonos skirtingo mato prekėms saugoti. Taip pat yra galimybė atkrauti didžiausias pakuotes, sujungti smulkias pakuotes į dideles ir t.t.[2]

Išanalizuotos sistemos turi visas reikalingas funkcijas kurios reikalingos, kad būtų pasiektas efektyvus sandėlio valdymas, tačiau norint įsidiesti ankščiau analizuojamas sistemas nedidelėms įmonėms kurios užsiima naudotų automobilio dalių prekyba tai kainuotų iš ties daug, kadangi yra reikalingas diegiamos sistemos pritaikymas prie įmonės veiklos. Tuo tarpu kuriama sandėlio valdymo sistema būtų šiuolaikiškas, patogus ir nebrangus įrankis skirtas būtent autolaužynų verslui. Klientai pradėję naudotis sistema netrukus galės įvertinti rezultatus – darbas taps kur kas efektyvesnis ir pelningesnis dėl sistemos teikiamų funkcijų:

- Su sistema visada žinosite, kiek ir kokių detalių turite sandėlyje.
- Detalių pardavimų procesą seksite realiu laiku.
- Visada galėsite tiksliai įvertinti, kiek laiko darbuotojai sugaišta automobilio ardymo darbams.
- Efektyvesnei ir pelningesnei veiklai nereikės naujų darbuotojų.
- Nereikės papildomų investicijų į techninę ar programinę įrangą.

2.2. PREKIŲ ŽYMĖJIMAS SANDĖLYJE

2.2.1. Prekių (detalių) žymėjimas

Norint greitai surasti reikiamą informaciją apie detalę esančią sandėlyje reikia jei suteikti kokį nors identifikavimo būdą, kadangi QR kodo šaknys yra būtent automobilių pramonė, tai ir kuriamoje sistemoje bus naudojamas būtent šis detalių identifikavimo būdas. Visų pirma tai būdas itin sparčiai, patikimai ir patogiai perduoti informaciją išmaniojo telefono savininkui, kurios tolimesnį panaudojimą jau sąlygoja tik informacijos pobūdis, kuris šiuo atveju bus detali informacija apie detalę.

QR kodas (Quick Response) – dvimačio brūkšninio kodo rūšis (angl. 2D matrix barcode), pastaruoju metu itin sparčiai plintanti įvairiausiose informacijos pateikimo bei perdavimo srityse. QR kodo šaknys – automobilių pramonėje, o tiksliau – šį kodą sukūrė japonų kompanija „Denso Wave“ (dukterinė Toyota kompanija) kaip tobulesnį pakaitalą dvimačiam brūkšniniam kodui, dar žinomam kaip UPC. Pagrindinė inovacijos savybė palyginus su UPC – į mažesnę plotą sutalpinamas gerokai didesnis informacijos kiekis, neprarandant kodo nuskaitymo spartos bei suteikiant galimybę klaidų kontrolei, o pirminė panaudojimo paskirtis buvo automobilinių dalių identifikavimas gamybos procese. Nors sukurtas ir Denso pradėtas naudoti dar 1994 metais, Japonijos Industrijos Asociacijos patvirtintas ir pradėtas naudoti 1997-aisiais, o 1999 metais paskelbtas žymėjimo standartu Japonijos industrinėje pramonėje, pirmuosius žingsnius į platųjį vartotojų ratą žengia dar tik keletą metų, kai 2006-aisiais buvo paskelbtas ISO standartu (ISO/IEC18004:2006).[4]

Didžiausias katalizatorius plintant QR kodui ne komerciniame ar industriniame sektoriuje, o tarp įprastų vartotojų – milžiniškais tempais populiarėjantys išmanieji telefonai, leidžiantys bet kuriam jų savininkui, pasinaudojant telefone įmontuota kamera bei papildoma programėle (kai kurie telefonai ją turi ir standartiškai), akimirksniu nuskaityti QR kode saugomą informaciją.[4]

QR kodo talpa

Įprastas UPC brūkšninis kodas talpina daugiausiai 30 simbolių, šiuo atveju QR galimybės – dešimtis ar net šimtus kartų didesnės: 7089 simboliai skaitiniams duomenims; [4]

- 7089 simboliai skaitiniams duomenims;
- 4296 simboliai mišriems skaitiniams/raidiniams duomenims (0–9, A–Z, tarpas, \$, %, *, +, -, ., /, :);
- 2953 Base256 (8 bitų) simboliai;
- 1817 Kanji/Kana simbolių.

Maksimalią kodo talpą lemia jo versija, kuri žymima skaičiumi nuo 1 iki 40. Pirmosios versijos kodo matmenys – 21×21 elementų, o versijos padidinimas vienetu yra lygus elementų padaugėjimui keturiais vienetais abiejomis kryptimis, t.y. 1 versija – 21×21 , 2 versija – 25×25 , 3 versija – 29×29 , ... , ir taip iki paskutiniosios, 40 versijos su 177×177 elementų. [4]

Klaidų korekcija

Taip pat maksimali kodo talpa priklauso ir nuo pasirinkto klaidų korekcijos lygio (L, M, Q, H – atitinkamai Low, Medium, Quality, High). Nuo pasirinkimo priklauso kokį procentą prarastų kodinių žodžių galima atstatyti (vienas kodinis žodis – 8 bitai): [4]

- L – leidžia atstatyti 7%;
- M – leidžia atstatyti 15%;
- Q – leidžia atstatyti 25%;
- H – leidžia atstatyti 30%.

Klaidų korekcijai naudojamas Reed-Solomon kodas, jį pridedant prie originalių duomenų. Korekcijos lygį atitinkantis procentas atitinka papildomai pridedamų duomenų procentą. Pvz. Jei koduojame 200 simbolių ir norime turėti galimybę 50 iš jų atstatyti praradimo atveju, kas sudaro 25% (Q lygis), turime pridėti papildomus 25% Reed-Solomon kodo duomenų, t.y. galutinis rezultatas bus 250 koduojamų simbolių. [4]

Kodo skanavimo aplinkybės

Dėl struktūros specifikos ir pagalbinių elementų QR kodas yra labai atsparus įvairiems iškreipymams ir skaitymo kampo nepastovumui. Plokštuma, ant kurios yra kodas, gali būti pasukta bet koku 360° kampu, nes pastarąjį skaitytuvas nustato pagal trijuose kampuose esančius nekintančius elementus (angl. finder pattern), kurių juodos-baltos spalvų santykis, nepriklausomai nuo žiūrėjimo kampo yra 1:1:3:1:1, kas padeda juos lengvai identifikuoti. Taip visus tris kampinius elementus jungia elementų eilė, angliškai vadinamų timing pattern, kuri yra skirta kodo bloko centrinės koordinatės nustatymui. Taip pat gali būti naudojamas (tačiau neprivalomas) elementas

alignment pattern, kuris padeda atlikti įvairių netiesinių iškraipymų korekcijas, pvz. kodo išgaubimą, įdubimą, ar perspektyvinius iškraipymus.[4]

2.2.2. Vartotojo sąsajos problemos

Norint sukurti intuityvę vartotojo sąsają, visų pirma reikia suprasti žmogų kaip individą – jis naudosis galutiniu produktu, ir bus sistemos sėkmės ar nesėkmės priežastis. Projektuojant vartotojo sąsają svarbu įvertinti tokius žmogaus bruožus ir savybes kaip suvokimą, regėjimą, dėmesingumą, atmintį ir kt.

Suvokimas ir jo įtaka panaudojamumui

Suvokimas nėra tikras atspindys to, kas yra pasaulyje. Žmogus suvokia tai, ką, bendru atveju, jis tikisi suvokti. Suvokimą lemia trys veiksniai:

- praeitis – žmogaus patirtis;
- dabartis – esamas kontekstas;
- ateitis – tikslai.

Praeities veiksnys rodo, kad žmogus, nuolat atliekantis tam tikrą veiksmą, tampa neatidus ir daro tai automatiškai. Iš šios savybės išplaukia vartotojo sąsajos nuoseklumo taisyklė – mygtukai, tam tikra informacija turėtų išlikti tose pačiose vietose skirtinguose sistemos languose. Dabartis rodo esamą kontekstą – tą patį žodį žmogus gali suprasti skirtingai, priklausomai nuo esamos situacijos.[6]

Ateitis siejasi su tikslais – žmogus, naudodamasis sistema, automatiškai nufiltruoja jam nesvarbius dalykus. Ieškant tam tikros informacijos, nesusiję su tikslais elementai nėra vien tik ignoruojami – dažnai jų paprasčiausiai net nepastebima.[6]

Šie žmogaus suvokimą lemiantys veiksniai turi poveikį vartotojo sąsajos panaudojamumui:

- **dviprasmybių vengimas:** svarbu vengti dviprasmiškos informacijos pateikimo, patikrinti, kad visi vartotojai interpretuotų rodomą vaizdą taip pat;[6]
- **nuoseklumo siekimas:** informacija ir valdymo elementai, atitinkantys tą pačią funkciją, turi būti pateikti toje pačioje vietoje skirtinguose sistemos languose, būti tos pačios spalvos, stiliaus, atspalvio ir t.t.;[6]
- **tikslų supratimas:** projektuotojai turi suprasti galimus vartotojo tikslus, įsitikinti, kad kiekviena sąveika su sistema aiškiai vestų prie galimo tikslo.[6]

Regėjimas ir jo įtaka panaudojamumui

Viena iš galimų teorijų apie regėjimo sistemą apima struktūrinį požiūrį – žmogus automatiškai nustato formas, figūras, objektus, o ne atskirus kampus, linijas. Ši teorija yra žinoma kaip regėjimo suvokimo Gestalto principai.[6]

Svarbiausi Gestalto principai yra artumas, panašumas, tęstinumas, uždarumas, simetrija, objektas-aplinka ir bendrumas:

Artumo principas teigia, kad santykinis atstumas tarp objektų veikia žmogaus suvokimą ir elementų skirstymą į grupes (1 pav.). Šis principas turi įtakos vartotojo sąsajos elementų išdėstymui: projektuotojai dažnai atskiria vartotojo sąsajos elementus naudodami skyriklių juostas, rėmelius, nors pagal artumo principą galima tiesiog naudoti didesnius tarpus. Tai sumažintų vartotojo sąsajos apkrovimą elementais, kodo dydį.[6]



1 pav. Artumo principo pavyzdys.

Panašumo principas teigia, kad panašūs objektai atrodo sugrupuoti (4 pav.). Informacinėse sistemose naudojamų formų laukai yra grupuojami panašumo principu. [6]



2 pav. Panašumo principo pavyzdys.

Tęstinumo principas teigia, kad žmogaus suvokimas remiasi ne atskirais segmentais, bet jų sujungimu (3 pav.). Šis principas yra pritaikomas vartotojo sąsajos slinkties juostose. [6]



3 pav. Tęstinumo principo pavyzdys.

Uždarumo principas teigia, kad žmogaus regėjimo sistema stengiasi sujungti atviras figūras taip, kad jos būtų suvokiamos kaip vienas objektas (4 pav.). Tai dažnai naudojama vartotojo sąsajos elementų vaizdavime, ikonose rodant vieną objektą ir už jo esančius kitų objektų kraštus, kurie suvokiami kaip vienas objektų rinkinys. [6]



4 pav. Uždarumo principo pavyzdys.

Objektas-aplinka principas teigia, kad žmogaus vaizduotė atskiria matomą vaizdą į objektą ir jo aplinką (5 pav.). Vartotojo sąsajoje šis principas naudojamas norint sudaryti išpūdį, kad objektas yra iškilęs virš aplinkos vaizdo, taip suteikiant jam daugiau dėmesio. [6]



5 pav. Objektas-aplinka principo pavyzdys.

Vienas iš svarbesnių vartotojo sąsajos projektavimo aspektų – struktūrinis informacijos pateikimas. Struktūrizuotas tekstas suteikia vartotojams galimybę greičiau rasti informaciją, palengvina skaitymą. Tai apima tiek didelius tekstus, tiek mažą informacijos kiekį (telefono numerius, sąskaitos kodus). Informacijos struktūrizavimo uždavinys yra pateikti vizualią hierarchiją, kuri suskirstytų informaciją į atskirus blokus, pažymėtų kiekvieną bloką tam, kad būtų lengviau jį identifikuoti, ir suteiktų svarbumo pojūtį.[7]

Kitas svarbus aspektas – spalvų pateikimas. Žmogaus akyje esanti tinklainė, kurioje yra fokusuojamas vaizdas, turi dviejų tipų receptorių ląsteles – stiebelius (skiriančius šviesos stiprumą) ir kūgelius (skiriančius spalvą). Kūgeliai yra trijų tipų ir skiria atitinkamai raudoną, žalią arba mėlyną spalvas. Šių receptorių siunčiamus signalus smegenys sujungia atimties principu ir leidžia matyti įvairias spalvas. Yra trys spalvų požymiai: tonas, šviesumas ir sodrumas. Kuriant vartotojo sąsają būtina atsižvelgti į žmogaus regėjimo subtilybes ir visus spalvų požymius. Pasirinkti harmoningą derinį padeda spalvų ratai, kuriuose braižant taisyklingas figūras gaunami derančių spalvų deriniai.[7]

2.3. MVC ARCHITEKTŪROS MODELIS

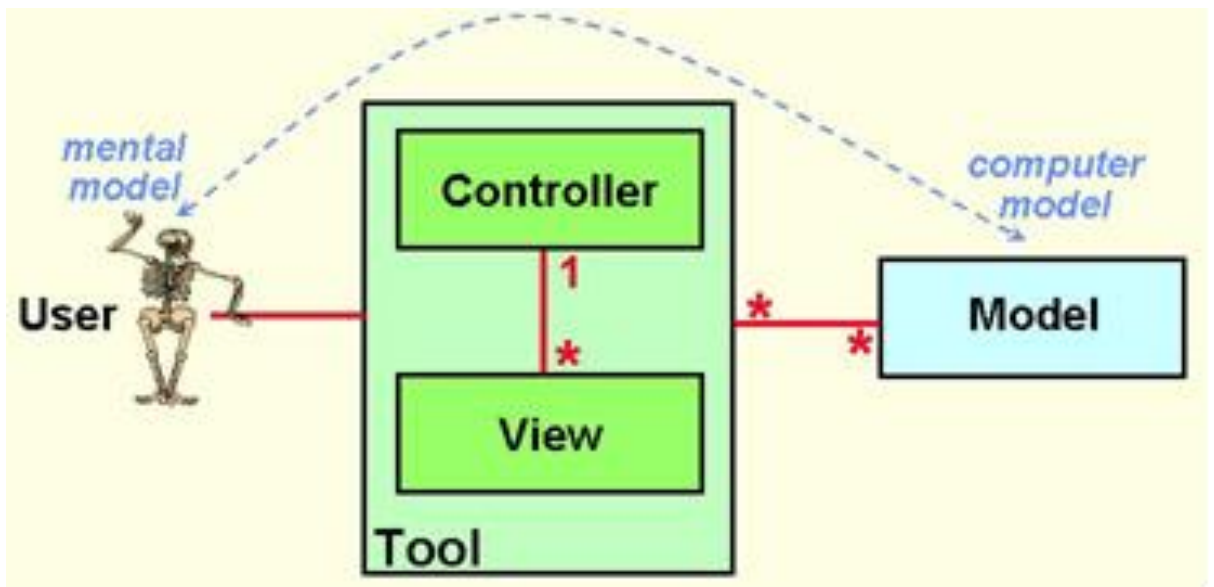
Model–view–controller (MVC) – tai programinės įrangos architektūra, šiuo metu laikoma architektūrine struktūra, naudojama projektuojant programinę įrangą. Struktūra izoliuoja „srities logiką“ (taikomoji logika vartotojui) nuo vartotojų sąsajos (įvesties ir pristatymo), leisdama nepriklausomai atlikti, testavimą ir palaikymą.[11]

MVC architektūros pradžia – kalba „Smalltalk“, kai 1979 m. tai buvo pirmą kartą pritaikyta programuotojo Trygve Reenskaug, siekiant gauti tradicinę įvestį, apdirbimą, ir išvestį su grafiniu vartotojų sąveikos modeliu. Tačiau, nesudėtinga pritaikyti šias sąvokas ir daugelyje kitų verslo aplikacijų sričių. MVC architektūra saityno aplikacijose taip pat naudojama seniai (šia architektūra yra paremti keli Apache projektai), o pats naudojimo bumus prasidėjo tada, kai O'Reilly apibrėžė terminą „Web 2.0“. Tada, paplitus „Ajax“ technologijų grupei, į interneto puslapį pradėta žiūrėti kaip į aplikaciją. Žinoma, tokioje situacijoje, norint suprogramuoti interneto svetainę, kaip aplikaciją reikia taikyti paprastų aplikacijų kūrimo filosofiją.[11]

MVC veikimas grindžiamas problemų atskyrimo (SoC) principu. Tokia projektavimo struktūra, kaip MVC, gali atskirti turinį nuo pristatymo ir duomenų apdorojimą (modelio) nuo turinio. Tai reiškia, kad programa išskaidoma į tokias dalis, kad jų funkcionalumas ir poreikiai nesidubliuotų. Šis principas taikomas daug kur. Vienas iš pavyzdžių XHTML ir CSS. XHTML kalbos paskirtis yra nurodyti dokumento struktūrą, CSS nurodo, kaip pats dokumentas turėtų atrodyti.[9]

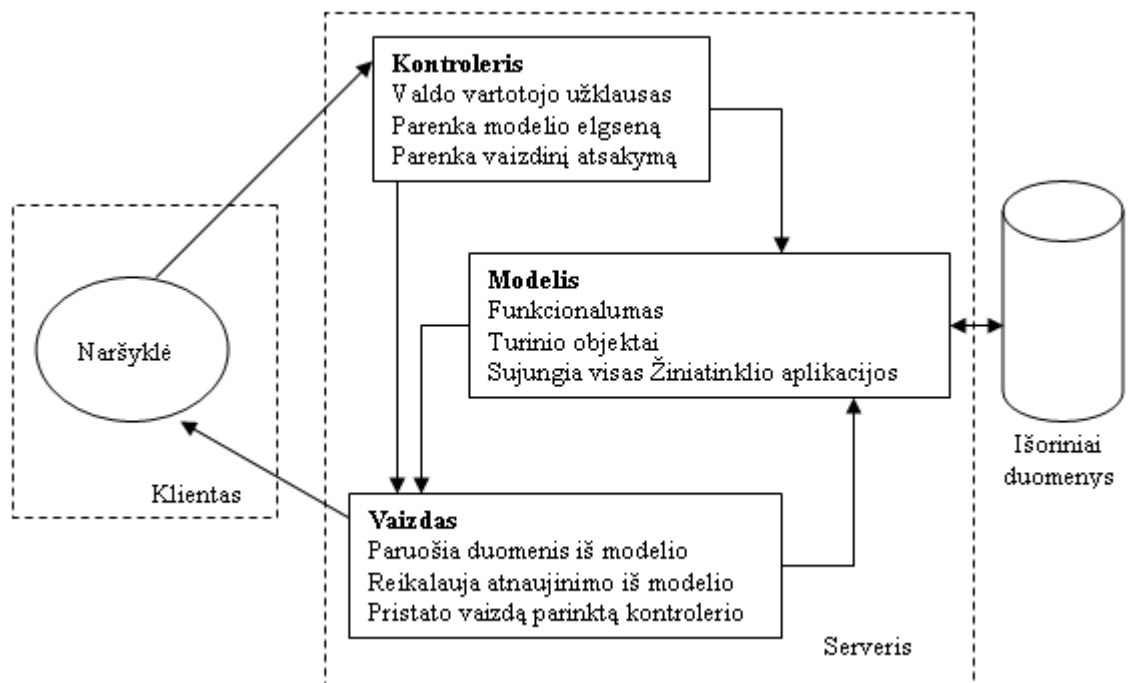
Objektiškai orientuotame programavime šis principas taikomas išskaidant užduočių atlikimą į objektus ir jų metodus. Pačiame MVC šis principas pasireiškia logikos, prezentacijos ir turinio atskyrimu. Kaip jau žinoma, MVC yra Model-View-Controller, – tai trys dalys, atsakingos už savo veiklos sritį.

Pagrindinis MVC tikslas – tiltu sujungti, protinį žmogaus kaip vartotojo modelį ir skaitmeninį modelį, kuris yra kompiuteryje (6 pav.). Struktūra yra naudinga, jei vartotojas turi pamatyti tą patį pavyzdinį elementą tuo pačiu laiku skirtinguose kontekstuose ir (arba) skirtingais požiūriais.[10]



6 pav. MVC ryšiai [10]

MVC buvo sugalvotas kaip bendras problemos sprendimas, kai vartotojai kontroliuoja didelę ir sudėtingą duomenų rinkinį. Tai yra plačiai paplitęs modelis, tarp daugelio programavimo kalbų ir struktūrų, kurių tikslas visiškai atskirti tris komponentus, bet kurioje saityno aplikacijoje (7 pav.):



7 pav. MVC veikimas.

Modelis – atstovauja įmonės duomenims ir verslo taisyklėms, kurios valdo šių duomenų prieigą ir atnaujinimus. Dažnai modelis naudojamas kaip programinės įrangos priartėjimui prie realiojo pasaulio proceso, kaip kad, paprasto realiojo pasaulio modeliavimo būdai prisitaiko, apibrėždami modelį. Modelis tai tam tikros srities duomenų valdymo ir pateikimo vienetas. Jis valdo elgesį ir taikomosios srities duomenis, atsako į užklausus dėl informacijos apie jos būseną (paprastai nuo vaizdo), ir atsako į instrukcijas pakeisti būseną (paprastai nuo valdiklio). Jis praneša stebėtojams

(paprastai vaizdui), kai informacija keičiasi, kad jie galėtų reaguoti. Pavyzdžiui: modelis gali būti atsakingas už mokesčius, privalo patikrinti, ar šiandien yra tam tikro vartotojo gimtadienis, išsiusti mėnesio naujienas el. paštu prenumeratoriams ir panašiai.[11]

Vaizdas – pateikia modelio turinius. Jis gauna prieigą prie įmonės duomenų per modelį ir apibrėžia, kaip tie duomenys turi būti pristatyti. Tai vaizdo atsakomybė palaikyti pastovumą, kai modelis keičiasi. Jis suteikia programos modeliui išvaizdą, tinkamą suprasti vartotojui (formas, sąrašai). Saityno aplikacijos tai ne kas kita kaip sugeneruotas HTML/CSS kodas. Vaizdas paverčia modelį į formą, tinkamą vartotojų sąveikai.[11]

Valdiklis – verčia sąveikas su vaizdu į veiksmus, kuriuos modelis įvykdys. Vartotojų sąveikos gali būti vykdomos mygtuko spragtelėjimu ar meniu pasirinkimais, saityno aplikacijoje, jie pasirodo, kaip GET ir POST HTTP prašymai. Veiksmai, įvykdyti modelio, apima verslo procesų formavimą arba modelio būsenos keitimą. Pagrįstas vartotojų sąveikomis ir modelio veiksmų rezultatais, valdiklis atsako, rinkdamas tinkamą vaizdą. Jis atsako už modelio valdymą ir vaizdų pateikimą pagal atitinkamus veiksmus (vartotojo interakciją): mygtuko paspaudimas, teksto įvedimas ir panašiai. Kai valdiklis gauna vartotojų įvestį jis pradeda daryti užklausas modelio objektams. Valdiklis priima įvestį iš vartotojo ir instruktuoja modelį bei vaizdą įvykdyti veiksmus, pagrįstus ta įvestimi.[11]

Naujokai programuotojai (ypač web programavimo), dažnai supranta MVC architektūros modelį visiškai neteisingai. Jie mano, kad modelis (model) - tai tik funkcijų ir (arba) metodų rinkinys, leidžiantis naudotis duomenimis, ir valdiklis (controller) – yra kaip sistemos dalis, kurioje yra tik verslo logika. Kaip rezultatas, – modelio kodas gaunamas kaip priemonė duomenims gauti iš duomenų bazės, o valdiklis yra tipiškas modulis, pilnas verslo logikos, arba tiesiog programa–interpretatorius.[10]

Taip suprasdami MVC, programuotojai pradėjo rašyti kodą, kuris yra dar žinomas kaip SKBV – Stori Kvaili Bjaurūs Valdikliai, kuriuos taip apibūdino iš Zend Framework programuotojų bendruomenės Padraic Brady (Fat Stupid Ugly Controllers). Todėl tampa akivaizdu, kad modelis MVC apibrėžime – tai ne tik duomenų prieigos kodas ir DBVS, bet ir srities logika. Savo ruožtu, valdikliai turi atsikratyti aplikacijos logikos (verslo logikos), kurią reikia perkelti į atskirą sluoksnį, pavyzdžiui, į paslaugų sluoksnį (Service Layer). Tokiu būdu valdiklis tampa „plonas“ ir atlieka tik sujungimo funkciją (glue layer) tarp atskirų komponentų sistemoje. [12]

MVC paprastai turi daug atmainų, tačiau esminis procesas vyksta taip:

1. Vartotojas atlieka tam tikrą veiksmą susijusį su GUI (pavyzdžiui paspaudžia mygtuką).
2. Valdiklis apdoroja šį veiksmą pagal nurodytą įvykių apdorojimo elementą (event handler).
3. Valdiklis iškviečia modelį, kuris, atlieka įvairius veiksmus su duomenimis (pavyzdžiui atnaujina registracijos informaciją, įdeda prekę į krepšelį ir panašiai).

4. Atitinkamas vaizdas grindžiamas modeliu, kad sugeneruotų atitinkamai naują vartotojo sąsają (pavyzdžiui, atvaizduoja naują registracijos informaciją, išmeta pranešimą apie prekę į krepšelį).
5. Vartotojo sąsaja laukia tolesnių veiksmų iš vartotojo.

MVC naudojamas tada, kai galima keletu skirtingų būdų peržiūrėti ir pasiekti reikiamus duomenis arba kai nežinomi būsimi reikalavimai duomenims peržiūrėti ir pasiekti. MVC struktūros naudojimo privalumai yra tie, kad ji leidžia keisti duomenis nepriklausomai nuo jų vaizdavimo būdo, ir atvirkščiai. Palaiko tų pačių duomenų vaizdavimą skirtingais būdais keičiant duomenis tik vienoje vietoje. Tačiau vienas iš MVC architektūros šablono trūkumų, tas kad gali prireikti sudėtingo programos kodo net ir tada, kai duomenų modelis bei sąveika tarp duomenų yra labai paprasta.[12]

2.4. IŠVADOS

1. Išanalizavus sandėlio valdymo sistemas matyti, kad visos jos turi reikalingas funkcijas tam, kad būtų pasiektas efektyvus sandėlio valdymas, tačiau norint įsidiegti tokias sistemas nedidelėms įmonėms kurios užsiima naudotų automobilio dalių prekyba tai kainuotų iš ties daug, kadangi yra reikalingas diegiamos sistemos pritaikymas prie įmonės veiklos. Tuo tarpu kuriama sandėlio valdymo sistema būtų šiuolaikiškas, patogus ir nebrangus įrankis skirtas būtent autolaužynų verslui.
2. Apžvelgus prekių žymėjimo sandėlyje galimybes buvo pasirinkta naudoti QR kodą – kadangi šis kodas iš pradžių buvo sukurtas naudoti būtent automobilių pramonėje ir buvo naudojamas detalių žymėjimui. QR kodu pažymėtos detalės informaciją greitai galima perskaityti su išmaniuoju įrengiu ir taip klientui pateikti jį dominančią informaciją.
3. Išanalizavus MVC architektūros privalumus buvo nuspręsta pasirinkti programavimo karkasą kuris yra paremtas MVC architektūros principu.

3. PROJEKTINĖ DALIS

3.1. PROJEKTO ĮGYVENDINIMO PLANAI

3.1.1. Darbų grafikas

2 lentelėje pateikiamas preliminarus projekto darbų įgyvendinimo grafikas.

1 lentelė. Projekto įgyvendinimo grafikas.

| Eil. nr. | Žingsnis | Terminas |
|----------|------------------------------|------------|
| 1 | Reikalavimų specifikavimas | 2016.03.04 |
| 2 | Architektūros specifikavimas | 2016.05.20 |
| 3 | Kuriamos PĮ realizacija | 2016.11.04 |
| 4 | PĮ testavimas | 2017.02.03 |

3.1.2. Projektavimo metodika

Projektavimui bus naudojama UML (Unified Modeling Language) – modeliavimo ir specifikacijų kūrimo kalba, skirta specifikuoti, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus. Kaip UML įrankis diagramoms braižyti bus naudojama „VisualParadigm“ programa, o visos programos kodas bus paremtas objektiškai orientuoto programavimo principais. Produktas bus kuriamas krioklio modeliu.

3.1.3. Programinė įranga

Planuojamos naudoti programinės įrangos sąrašas yra pateiktas 2 lentelėje.

2 lentelė. Programinės įrangos sąrašas.

| Eil. Nr. | Resursas | Kaina |
|-----------------|----------------|---------------------------|
| 1 | PHPStorm | 200 Eur |
| 2 | Brackets | 0 Eur (įrankis nemokamas) |
| 3 | WAMP | 0 Eur (įrankis nemokamas) |
| 4 | Composer | 0 Eur (įrankis nemokamas) |
| 5 | VisualParadigm | 300 Eur |
| 6 | Windows 10 x64 | 200 Eur |
| Iš viso: | | 700 Eur |

3.1.4. Biuro įranga

Darbui reikalinga biuro įranga pateikta 3 lentelėje.

3 lentelė. Biuro įrangos sąrašas.

| Eil. nr. | Resursas | Kaina |
|-----------------|--------------------|-----------------|
| 1 | Patalpos (18 mėn.) | 4000 Eur. |
| 2 | Kompiuteris | 900 Eur. |
| 3 | Spausdintuvas | 150 Eur. |
| 4 | Raštinės reikmenys | 100 Eur. |
| Iš viso: | | 5150 Eur |

3.1.5. Projekto vykdytojai

Projektui reikalingi šie gyvieji resursai:

- **Projekto vadovas/analitikas** – būtų atsakingas už projekto proceso valdymą, darbų, resursų paskirstymą, projekto kūrimo eigos stebėjimą, reikalavimų surinkimą.
- **Projektuotojas** – atsakingas už galutinį projekto architektūros modelį.
- **Dizaineris/koduotojas** – atsakingas už sistemos dizaino kūrimo darbus.
- **Programuotojas** – atsakingas už pagrindinius programavimo darbus.
- **Programuotojas/testuotojas** – atsakingas už programavimo darbus ir sukurtos sistemos ištestavimą.

Programinės įrangos kūrimui reikalingi darbuotojai yra pateikti 4 lentelėje.

4 lentelė. Gyvųjų resursų sąrašas.

| Eil. nr. | Resursas | Kaina (1 mėn.) x Laikotarpis |
|-----------------|-----------------------------|------------------------------|
| 1 | Projekto vadovas/analitikas | 800 Eur.* 18 |
| 2 | Projektuotojas | 900 Eur. * 3 |
| 3 | Dizaineris/koduotojas | 700 Eur.* 4 |
| 4 | Programuotojas | 800 Eur.* 11 |
| 5 | Programuotojas/testuotojas | 700 Eur.* 6 |
| Iš viso: | | 32500 Eur |

3.1.6. Biudžetas

Planuojamas programinės įrangos biudžetas pateiktas 5 lentelėje.

5 lentelė. Planuojamas biudžetas.

| Eil. nr. | Resursas | Kaina |
|-----------------|----------------------|-----------------------------|
| 1 | Programinė įranga | 1000 Eur. |
| 2 | Biuro įranga | 0 Eur. (aprūpina užsakovas) |
| 3 | Gyvieji resursai | 35000 Eur. |
| 4 | Nenumatytos išlaidos | 600 Eur. |
| Iš viso: | | 36600 Eur |

3.1.7. Kokybės vertinimas

Programinės įrangos kokybę vertinsime remdamiesi kriterijais kurie yra pateikti 6 lentelėje.

6 lentelė. Programinės įrangos vertinimo kriterijai.

| Eil. nr. | Pavadinimas | Apibūdinimas |
|----------|-------------------------------|---|
| 1 | Sąskaitų generavimas | Galimybė generuoti sąskaitas faktūras. |
| 2 | Vartotojo sąsaja | Suprantama sąsaja šios srities specialistams. |
| 3 | Stabilumas | Sistemos veikimo laikas be klaidų. |
| 4 | Mokama/Nemokama | Ar yra mokama/nemokama versijos. |
| 5 | Cloud based (veikia debesyje) | Galimybė naudotis sistema iš, bet kurios darbo vietos kur tik yra internetas. |

7 lentelė. Panašaus pobūdžio programų vertinimas pagal 6 lentelėje pateiktus kriterijus.

| Eil. nr. | Pavadinimas | EquinosEurope Vision | Picker | Cadence WMS | ProVision |
|----------|-------------------------------|----------------------|--------|-------------|-----------|
| 1 | Sąskaitų generavimas | + | + | - | + |
| 2 | Vartotojo sąsaja | + | + | - | + |
| 3 | Stabilumas | + | + | + | - |
| 4 | Mokama/Nemokama | +/- | +/- | +/- | +/- |
| 5 | Cloud based (veikia debesyje) | + | + | - | - |

3.2. SISTEMOS APIBŪDINIMAS

3.2.1. Programų sistemos funkcijos

Pagrindinis darbo tikslas sukurti specializuotą sandėlio valdymo sistemą kuri būtų skirta naudotomis automobilių dalimis prekiaujančioms įmonėms ir veiktų saityne. Ši sistema pasižymės tuo, kad bus paprasta naudoti, nereikalaus jokių ypatingų kompiuterinių žinių iš jos vartotojų.

Pagrindiniame lange pateikiami logiškai išdėstyti meniu mygtukai (įvedimas, paieška, užsakymai, pardavimai, ataskaitos) kurie leidžia rasti ieškomas funkcijas. Sistema leis darbuotojams į sistemą įvesti informaciją apie sandėlyje esančias detales, įtrauktai detalei bus suteikiamas identifikacinis numeris. Jei detalė dar nebus išmontuota, sistemoje galima įvesti užsakymą išmontavimui ir kliento kontaktinius duomenis, kad išėmus reikiamą detalę iš automobilio būtų

galimybė susisiekti su užsakovu. Gavę kliento užklausą darbuotojai pasinaudos paieškos funkcija ir greitai sužinos ar sandėlyje yra reikalinga detalė. Tuo atveju jei klientas dar nėra 100% apsisprendęs dėl pirkimo, detalę sistemoje jam galima rezervuoti tam tikram laikui, kai detalė parduodama, darbuotojas vienu mygtuko paspaudimu pažymės, kad prekės jau nebėra. Jei vienodų detalių sistemoje bus daugiau nei viena po pardavimo tiesiog sumažės jų skaičius. Darbuotojai sugeneravę pardavimų ataskaitas galės įvertinti, kurios detalės pelningiausios, kokios prekės parduodamos greičiausiai, o kurios ilgai užsilaiko sandėlyje. Kitas ataskaitos tipas parodys, kiek detalių buvo įvesta tam tikrą dieną. Duomenų įvedimas yra priskiriamas konkrečiam darbuotojui.

3.2.2. Sistemos kontekstas

Sukurta sistema veiks atskirai ir nepriklausys nuo kitų sistemų.

3.2.3. Vartotojo charakteristikos

Būsimos sistemos vartotojai bus sandėlininkai ir kiti analizuojamos srities atstovai. Todėl galima įvardinti tokias pagrindines darbuotojų charakteristikas:

- Geri darbo kompiuteriu įgūdžiai;
- Sandėlio valdymo procesų išmanymas;
- Automobilio ir jo dalių išmanymas

3.2.4. Vartotojo problemos

Svarbiausios vartotojų problemos, kurias turi padėti spręsti kuriama sistema tai yra greitesnis ir tikslesnis informacijos suteikimas klientui apie jį dominančią detalę, bei greitesnis tos detalės suradimas sandėlyje ir jos pateikimas klientui.

3.2.5. Vartotojo tikslai

Pagrindinis šią sistemą naudojančio vartotojo tikslas yra turėti tvarkingus sandėlius ir visada žinoti tikslią informaciją kiek, kur ir kokių detalių yra jo įmonės sandėlyje.

3.2.6. Bendri apribojimai

- Kompiuteryje įdiegta operacinė sistema ir interneto naršyklė
- Nuolatinis interneto ryšys kompiuteryje.

3.3. SISTEMOS PASKIRTIS

Daugelis šiomis dienomis veikiančių įmonių ir organizacijų, kurių veikloje valdomi pakankamai dideli prekių srautai, naudoja vienokias ar kitokias kompiuterines apskaitos sistemas. Be jų jau neįsivaizduojama jokia rimtesnė buhalterinė apskaita ar prekių valdymas sandėlyje, programų naudingumas įmonėms jau seniai nebekelia jokių abejonių. Sėkmingai įmonėje įdiegta sandėlio valdymo sistema leidžia ženkliai, padidinti darbo našumą visoje įmonėje.

Dalinai reikalavimus sistemai gali atitikti jau egzistuojantys rinkoje komerciniai produktai. Tačiau, sistemoje egzistuoja ir daug reikalavimų kurie yra labai specifiniai ir joki iki šiol rinkoje egzistuojantys produktai jų neatitinka. Todėl čia neįmanoma apsieiti be specializuotos programinės įrangos, sukurtos atlikti konkrečias reikiamas funkcijas.

Kita svarbi priežastis yra produkto kaina. Nedidelės įmonės negali sau leisti užsisakyti brangų komercinį produktą. Kaip žinoma dauguma įmonių prekiaujančių naudotomis automobilių dalimis yra priskiriamos prie smulkaus ar vidutinio verslo. Kaina taip pat įtakoja ir realizavimo priemonių, tokių kaip duomenų bazės pasirinkimą.

Būtent dėl šių priežasčių, bus pradėta projektuoti nauja sistema.

3.4. SISTEMOS TIKSLAI

Sistema turi vieną pagrindinį tikslą: padėti įmonėms užsiimančioms naudotų automobilių dalių prekyba lengviau ir efektyviau valdyti savo sandėlių veiklą.

3.5. IŠVADOS

1. Sistemos veikimas nepriklausys nuo kitų informacinių sistemų.
2. Sistema orientuota į mažas įmones kurios prekiauja naudotomis automobilių dalimis, nes jos kūrimo ir priežiūros kaina yra nedidelė.

4. REIKALAVIMŲ SPECIFIKACIJA

4.1. UŽSAKOVAS IR SISTEMOS VARTOTOJAI

4.1.1. Užsakovas

Užsakovas yra Kauno Technologijos Universiteto, Programų inžinerijos katedros profesorius Vacius Jusas. Ši katedra buvo įkurta 2001 metais pertvarkius Programinės įrangos katedrą į Programų inžinerijos, Kompiuterinių tinklų ir Sisteminės analizės katedras.

4.1.2. Vartotojai

Vartotojo kategorija: Administratorius;

Vartotojo sprendžiami uždaviniai: vartotojas, kurio darbas kurti ir administruoti kitus vartotojus, priimti užsakymus iš klientų ir juos registruoti sistemoje;

Patirtis dalykinėje srityje: Patyręs;

Patirtis informacinėse technologijose: naujokas;

Papildomos vartotojo charakteristikos: nėra;

Vartotojų prioritetai: svarbiausi vartotojai.

Vartotojo kategorija: Sandėlininkas;

Vartotojo sprendžiami uždaviniai: tai vartotojas kuris kuria sistemos katalogo struktūrą, prideda į jį naujas dalis ar jas pašalina, vartotojas taip pat atlieka pardavėjo vaidmenį;

Patirtis dalykinėje srityje: įprastas darbuotojas;

Patirtis informacinėse technologijose: naujokas;

Papildomos vartotojo charakteristikos: nėra;

Vartotojų prioritetai: antraeiliai vartotojai.

4.2. ĮPAREIGOJANTYS APRIBOJIMAI

4.2.1. Apribojimai sprendimui

- Sistema turi veikti per interneto naršyklę;

4.2.2. Diegimo aplinka

- Pagrindinė sistemos dalis įdiegiama į turimą užsakovo serverį;
- Vartotojų kompiuteriuose klientinės dalies nėra, nes viskas vyksta per interneto naršyklę;
- Papildoma sistemos dalis – aplikacija įdiegiama į sandėlininko telefoną.

4.2.3. Bendradarbiaujančios sistemos

- Bendraujančių sistemų nenumatyta.

4.2.4. Komerciniai specializuoti programų paketai

- Kuriant šią sistemą nenumatoma naudoti jokių komercinių specializuotų paketų.

4.2.5. Numatoma darbo vietos aplinka

- Darbo vieta remsis interneto naršykle, todėl būtina užtikrinti pastovų ryšį su serveriu, kadangi kitaip sistema neveiks.

4.2.6. Sistemos kūrimo terminai

Pagrindiniai projekto etapai ir jų atlikimo terminai:

- Projekto darbų plano sudarymas (2016-01-04)
- Projekto reikalavimų specifikacijos sudarymas (2016-03-10)
- Projekto architektūros specifikacijos sudarymas (2016-04-21)
- Atliekami sistemos programavimo darbai (2016-09-29)
- Sistemos testavimo plano sudarymas ir jo vykdymas (2016-11-24)
- Sistemos naudotojo dokumentacijos sudarymas (2016-12-22)

4.2.7. Sistemos kūrimo biudžetas

Projektui realizuoti reikalingi resursai pateikti 8 lentelėje

8 lentelė. Projektui realizuoti reikalingi resursai

| NR. | Darbuotojo pareigos | Valandinis įkainis (EUR) | Laikas (Val) | Bendra suma (EUR) |
|-----------------|-----------------------------|--------------------------|--------------|-------------------|
| 1 | Projekto vadovas/analitikas | 20 | 120 | 2400 |
| 2 | Dizaineris/koduotojas | 15 | 40 | 600 |
| 3 | Programuotojas | 25 | 160 | 4000 |
| 4 | Programuotojas/testuotojas | 20 | 100 | 2000 |
| Iš viso: | | 80 | 420 | 9000 |

Pastaba: Į kaina neįskaičiuota nei aparatūros nei papildomos programinės įrangos kaina. Kadangi įmonė verčiasi ir kitais projektais, tai programuotojai jau turi visą aparatūrą bei programinę įrangą sukurti šiai sandėlio valdymo sistemai.

4.3. SVARBŪS FAKTAI IR PRIELAIIDOS

4.3.1. Svarbūs faktai

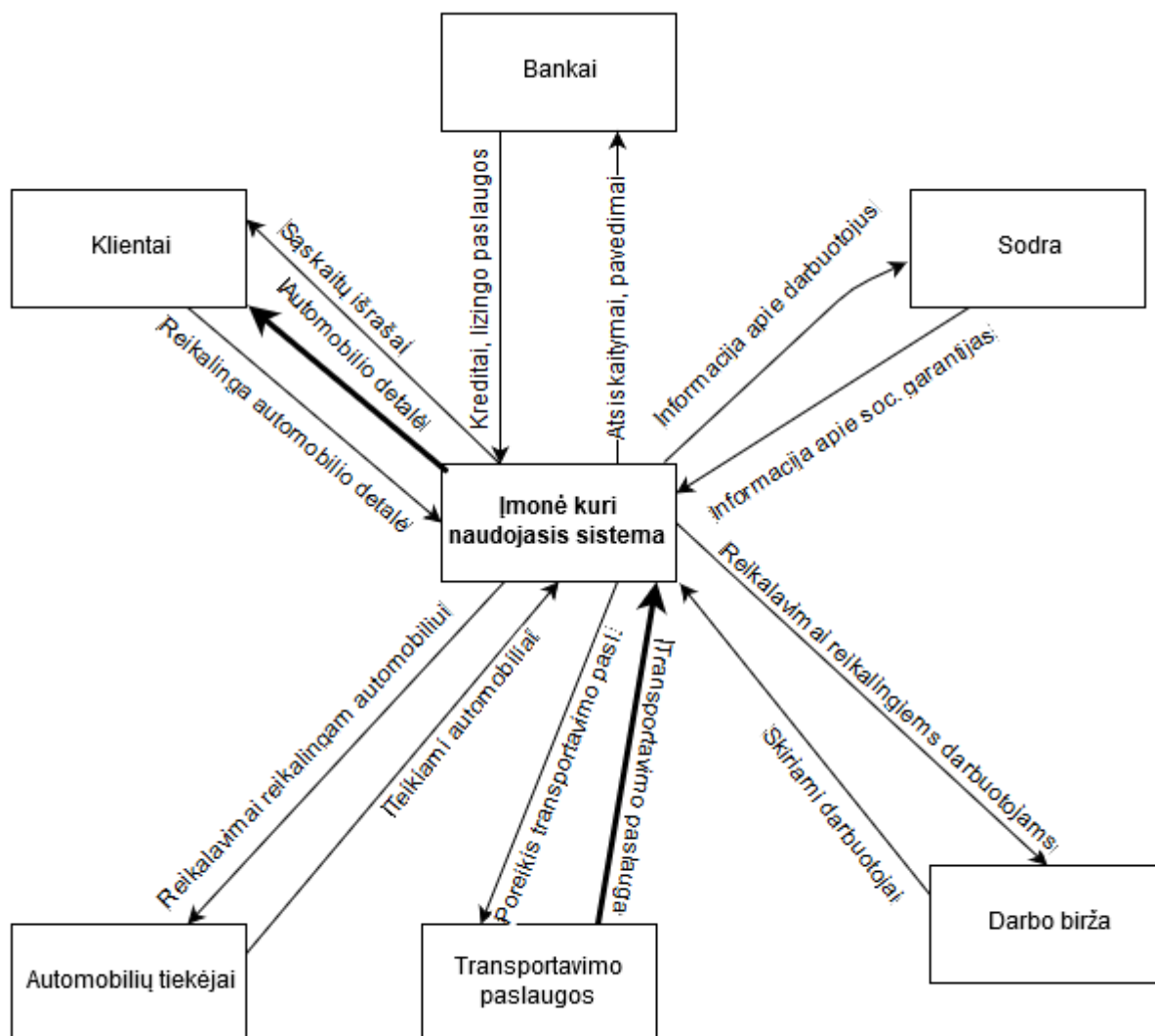
- Kuriama programinė įranga bus pateikta kaip atviro kodo programinė įranga visiems norintiems ja naudotis.

4.3.2. Prielaidos

- Dalis sistemos funkcijų bus realizuotos pasinaudojus jau sukurtais kitų šalių komponentais.

4.4. VEIKLOS SUDĖTIS

4.4.1. Veiklos kontekstas



8 pav. Aukščiausiojo lygio kontekstinė DFD

4.4.2. Veiklos padalijimas

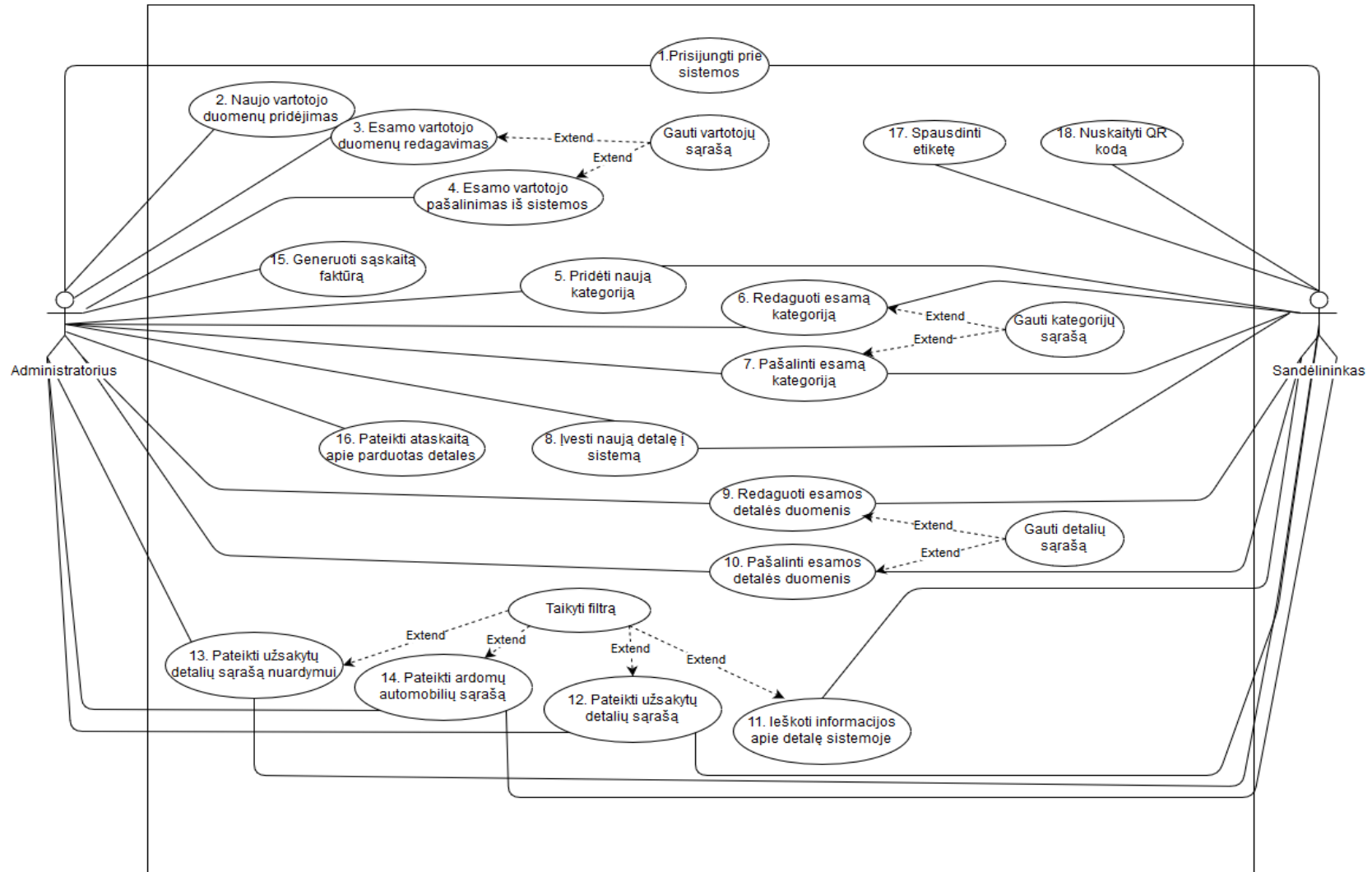
9 lentelė. Veiklos padalijimas.

| Eil. Nr. | Įvykio pavadinimas | Įeinantys / išeinantys informacijos srautai |
|----------|--|---|
| 1 | Naujo darbuotojo įregistravimas | Darbuotojų duomenys (in) |
| 2 | Gautas naujas automobilis ardymui | Automobilio duomenys (in) |
| 3 | Kliento užsakymas konkrečiai detalei | Užsakymo duomenys (in) |
| 4 | Klientas rezervuoja norimą detalę | Rezervavimo duomenys (in) |
| 5 | Darbuotojai gauna informaciją apie reikalingas detales | Duomenų teikimas (out) |

| | | |
|---|---|-----------------|
| 6 | Darbuotojas peržiūri leistinas ataskaitas | Ataskaita (out) |
| 7 | Įmonės vadovas gauna ataskaitas apie parduotas detales | Ataskaitos (in) |
| 8 | Darbuotojai spausdina etiketes klijuojamas ant detalių | Etiketės (out) |
| 9 | Pardavėjas pagal kliento pageidavimą pateikia jam sąskaitą faktūrą. | Sąskaitos (out) |

4.5. SISTEMOS SUDĖTIS

4.5.1. Sistemos ribos



9 pav. Sistemos panaudos atvejai.

4.5.2. Panaudojimo atvejų sąrašas

10 lentelė. PA prisijungimas prie sistemos (vartotojų autentifikavimas).

| 1 panaudojimo atvejis: | Prisijungimas prie sistemos (vartotojų autentifikavimas). |
|-------------------------|--|
| Aktoriai: | Administratorius, sandėlininkas |
| Prieš sąlyga: | Norima prisijungti prie valdymo posistemės. |
| Sužadinimo sąlyga: | Vartotojas reikia pasinaudoti valdymo posisteme. |
| Po sąlyga: | Vartotojas autentifikuojamas ir gali naudotis valdymo posisteme |
| Pagrindinis scenarijus: | 1. Vartotojas suveda vartotojo vardą ir slaptažodį; 2. Sistema patikrina duomenis; 3. Vartotojas prijungiamas prie sistemos. |

11 lentelė. PA naujo vartotojo duomenų pridėjimas į sistemą.

| 2 panaudojimo atvejis: | Naujo vartotojo duomenų pridėjimas į sistemą. |
|----------------------------|--|
| Tikslas: | Pridėti/užregistruoti naują sandėlio darbuotoją sistemoje. |
| Aktoriai: | Administratorius |
| Nefunkciniai reikalavimai: | Turi būti užpildyti privalomi laukai. Identifikavimo vardai ir kiti kritiniai laukai negali kartotis. |
| Prieš sąlyga: | Administratorius prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Atsirado poreikis pridėti naują vartotoją. |
| Po sąlyga: | Pridedamas naujas vartotojas. |
| Pagrindinis scenarijus: | 1. Administratorius suveda vartotojo informaciją; 2. Sistema patikrina duomenis; 3. Sukuriamas vartotojas. |
| Alternatyvūs scenarijai: | Esant neteisingiems duomenims, sistema siūlo patikrinti duomenis ir bandyti dar kartą. |

12 lentelė. PA esamo vartotojo duomenų redagavimas.

| 3 panaudojimo atvejis: | Esamo vartotojo duomenų redagavimas. |
|----------------------------|---|
| Tikslas: | Pakeisti esamo vartotojo informaciją. |
| Aktoriai: | Administratorius |
| Nefunkciniai reikalavimai: | Turi būti užpildyti privalomi laukai. Identifikavimo vardai ir kiti kritiniai laukai negali kartotis. |
| Prieš sąlyga: | Administratorius prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Pasikeitė esamo vartotojo duomenys. |
| Po sąlyga: | Vartotojo duomenys atnaujinami. |
| Pagrindinis scenarijus: | 1. Administratorius pakeičia esamo vartotojo duomenis; 2. Sistema patikrina duomenis; 3. Vartotojo informacija atnaujinama. |
| Alternatyvūs scenarijai: | Esant neteisingiems duomenims, sistema siūlo patikrinti duomenis ir bandyti dar kartą. |

13 lentelė. PA esamo vartotojo pašalinimas iš sistemos.

| 4 panaudojimo atvejis: | Esamo vartotojo pašalinimas iš sistemos. |
|------------------------|---|
| Tikslas: | Pašalinti sistemoje esantį vartotoją. |
| Aktoriai: | Administratorius |
| Prieš sąlyga: | Administratorius prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Vartotojo paskyra nebeaktuali. |

| | |
|--------------------------|--|
| Po sąlyga: | Vartotojas pašalinamas iš sistemos. |
| Pagrindinis scenarijus: | 1. Administratorius iš sąrašo pasirenka reikiamą vartotoją; 2. Šalinimas. 3. Parodomas rezultatas. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

14 lentelė. PA pridėti naują kategoriją.

| 5 panaudojimo atvejis: | Pridėti naują kategoriją. |
|-------------------------------|--|
| Tikslas: | Pridėti naują kategoriją sistemoje. |
| Aktoriai: | Administratorius |
| Nefunkciniai reikalavimai: | Turi būti užpildyti privalomi laukai. Identifikavimo vardai ir kiti kritiniai laukai negali kartotis. |
| Prieš sąlyga: | Administratorius prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Naujos kategorijos poreikis. |
| Po sąlyga: | Sistemoje pridedama nauja kategorija. |
| Pagrindinis scenarijus: | 1. Administratorius suveda naujos kategorijos informaciją; 2. Sistema patikrina duomenis; 3. Sukuriama nauja kategorija. |
| Alternatyvūs scenarijai: | Esant neteisingiems duomenims, sistema siūlo patikrinti duomenis ir bandyti dar kartą. |

15 lentelė. PA redaguoti esamą kategoriją.

| 6 panaudojimo atvejis: | Redaguoti esamą kategoriją. |
|-------------------------------|---|
| Tikslas: | Redaguoti esamą kategoriją. |
| Aktoriai: | Administratorius |
| Nefunkciniai reikalavimai: | Turi būti užpildyti privalomi laukai. Identifikavimo vardai ir kiti kritiniai laukai negali kartotis. |
| Prieš sąlyga: | Administratorius prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Reikalinga pakeisti esamos kategorijos duomenis. |
| Po sąlyga: | Atnaujinami kategorijos duomenys. |
| Pagrindinis scenarijus: | 1. Administratorius pakeičia kategorijos informaciją; 2. Sistema patikrina duomenis; 3. Kategorijos duomenys atnaujinami. |
| Alternatyvūs scenarijai: | Esant neteisingiems duomenims, sistema siūlo patikrinti duomenis ir bandyti dar kartą. |

16 lentelė. PA pašalinti esamą kategoriją.

| 7 panaudojimo atvejis: | Pašalinti esamą kategoriją. |
|-------------------------------|--|
| Tikslas: | Pašalinti esamą kategoriją. |
| Aktoriai: | Administratorius |
| Prieš sąlyga: | Administratorius prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Kategorija tampa nebeaktuali. |
| Po sąlyga: | Kategorija pašalinama. |
| Pagrindinis scenarijus: | 1. Iš sąrašo pasirenkama reikiama kategorija; 2. Šalinimas. 3. Parodomas rezultatas. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

17 lentelė. PA įvesti naują detalę į sistemą.

| 8 panaudojimo atvejis: | Įvesti naują detalę į sistemą. |
|-------------------------------|---|
| Tikslas: | Papildyti katalogą naujomis detalėmis. |
| Aktoriai: | Sandėlininkas |
| Prieš sąlyga: | Sandėlininkas prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Atsiranda sistemoje neįvestų detalių. |
| Po sąlyga: | Detalė pridedama į sistemą. |
| Pagrindinis scenarijus: | 1. Sandėlininkas suveda naujos detalės informaciją; 2. Sistema patikrina duomenis; 3. Nauja detalė pridedama į sistemą. |
| Alternatyvūs scenarijai: | Esant neteisingiems duomenims, sistema siūlo patikrinti duomenis ir bandyti dar kartą. |

18 lentelė. PA redaguoti esamos detalės duomenis.

| 9 panaudojimo atvejis: | Redaguoti esamos detalės duomenis. |
|-------------------------------|---|
| Tikslas: | Redaguoti esamos detalės duomenis. |
| Aktoriai: | Sandėlininkas |
| Prieš sąlyga: | Sandėlininkas prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Atsiranda poreikis pakeisti duomenis apie detalę. |
| Po sąlyga: | Detalės duomenys pakeičiami. |
| Pagrindinis scenarijus: | 1. Sandėlininkas pakeičia detalės duomenis; 2. Sistema patikrina duomenis; 3. Detalės duomenys atnaujinami. |
| Alternatyvūs scenarijai: | Esant neteisingiems duomenims, sistema siūlo patikrinti duomenis ir bandyti dar kartą. |

19 lentelė. PA pašalinti esamą detalę iš sistemos.

| 10 panaudojimo atvejis: | Pašalinti esamą detalę iš sistemos. |
|--------------------------------|--|
| Tikslas: | Pašalinti detalę iš sistemos. |
| Aktoriai: | Sandėlininkas |
| Prieš sąlyga: | Sandėlininkas prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Detalė tampa nebeaktuali. |
| Po sąlyga: | Detalės duomenys pašalinami iš sistemos. |
| Pagrindinis scenarijus: | 1. Iš sąrašo pasirenkama reikiama detalė; 2. Šalinimas. 3. Parodomas rezultatas. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

20 lentelė. PA ieškoti informacijos apie detalę sistemoje.

| 11 panaudojimo atvejis: | Ieškoti informacijos apie detalę sistemoje. |
|--------------------------------|--|
| Tikslas: | Gauti informaciją apie detalės buvimą. |
| Aktoriai: | Administratorius/Sandėlininkas |
| Prieš sąlyga: | Vartotojas prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Klientas pateikia užklausą. |
| Po sąlyga: | Gaunama informacija apie ieškomą detalę. |
| Pagrindinis scenarijus: | 1. Įvedama ieškomos detalės pavadinimas; 2. Ieškojimas. |

| | |
|--------------------------------|---|
| 11 panaudojimo atvejis: | Ieškoti informacijos apie detalę sistemoje. |
| | 3. Parodomas rezultatas. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

21 lentelė. PA pateikti užsakytų detalių sąrašą.

| | |
|--------------------------------|---|
| 12 panaudojimo atvejis: | Pateikti užsakytų detalių sąrašą. |
| Tikslas: | Peržiūrėti užsakytų detalių sąrašą. |
| Aktoriai: | Administratorius/Sandėlininkas |
| Prieš sąlyga: | Vartotojas prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Reikalinga informacija apie užsakytas detales. |
| Po sąlyga: | Gaunama informacija apie užsakytas detales. |
| Pagrindinis scenarijus: | 1. Pasirenkamas užsakymų laikotarpis. 2. Parodomas rezultatas. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

22 lentelė. PA pateikti užsakytų detalių sąrašą nuardymui.

| | |
|--------------------------------|---|
| 13 panaudojimo atvejis: | Pateikti užsakytų detalių sąrašą nuardymui. |
| Tikslas: | Gauti informaciją apie šiuo metu klientui reikalingas detales. |
| Aktoriai: | Administratorius/Sandėlininkas |
| Prieš sąlyga: | Vartotojas prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Reikalinga informacija apie detales užsakytas nuardymui. |
| Po sąlyga: | Gaunama informacija apie detales kurias reikia nuardyti. |
| Pagrindinis scenarijus: | 1. Sistemoje pasirenkama skiltis ardomi automobiliai 2. Pasirenkama sub skiltis laukia nuardymo. 2. Parodomas rezultatas. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

23 lentelė. PA pateikti ardomų automobilių sąrašą.

| | |
|--------------------------------|---|
| 14 panaudojimo atvejis: | Pateikti ardomų automobilių sąrašą. |
| Tikslas: | Gauti informaciją apie šiuo metu ardomus automobilius. |
| Aktoriai: | Administratorius/Sandėlininkas |
| Prieš sąlyga: | Sandėlininkas prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Reikalinga informacija apie ardomus automobilius. |
| Po sąlyga: | Gaunama informacija apie ardomus automobilius. |
| Pagrindinis scenarijus: | 1. Sistemoje pasirenkama skiltis ardomi automobiliai; 2. Parodomas rezultatas. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

24 lentelė. PA generuoti sąskaitą faktūrą.

| | |
|--------------------------------|---|
| 15 panaudojimo atvejis: | Generuoti sąskaitą faktūrą. |
| Tikslas: | Sugeneruoti sąskaitą faktūrą. |
| Aktoriai: | Administratorius(pardavėjas) |
| Prieš sąlyga: | Administratorius prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Klientui reikalinga sąskaita faktūra. |
| Po sąlyga: | Sistema sugeneruoja sąskaitą faktūrą. |
| Pagrindinis scenarijus: | 1. Administratorius pasirenka generuoti sąskaitą faktūrą; 2. Sugeneruojama sąskaita faktūra. |

25 lentelė. PA pateikti ataskaitą apie parduotas detales.

| 16 panaudojimo atvejis: | Pateikti ataskaitą apie parduotas detales. |
|--------------------------|--|
| Tikslas: | Gauti ataskaitą apie parduotas detales. |
| Aktoriai: | Administratorius |
| Prieš sąlyga: | Administratorius prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Reikalinga gauti parduotų detalių ataskaitą. |
| Po sąlyga: | Pateikiama sugeneruota ataskaita. |
| Pagrindinis scenarijus: | 1. Administratorius pasirenka laikotarpį už kurį nori gauti ataskaitą; 2. Parodomas rezultatas. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

26 lentelė. PA spausdinti etiketę.

| 17 panaudojimo atvejis: | Spausdinti etiketę. |
|--------------------------|---|
| Tikslas: | Gauti etiketę su QR kodu. |
| Aktoriai: | Sandėlininkas |
| Prieš sąlyga: | Sandėlininkas prisijungęs prie valdymo posistemės. |
| Sužadinimo sąlyga: | Reikalinga sandėliuoti naujai gautą detalę. |
| Po sąlyga: | Sugeneruojama ir atspausdinama etiketė. |
| Pagrindinis scenarijus: | 1. Sandėlininkas įveda duomenis į sistemą apie naujai gautą detalę; 2. Pasirenka spausdinti etiketę; 3. Atspausdinama etiketė klijuojama ant detalės. |
| Alternatyvūs scenarijai: | Esant klaidai išvedamas apie tai informuojantis pranešimas. |

27 lentelė. Nuskaityti QR kodą.

| 18 panaudojimo atvejis: | Nuskaityti QR kodą. |
|--------------------------|---|
| Tikslas: | Nuskaičius QR kodą mobiliuoju telefonu gauti detalią informaciją apie detalę. |
| Aktoriai: | Sandėlininkas |
| Prieš sąlyga: | Sandėlininkas turi mobilų telefoną kuriame įdiegta kuriamos sistemos modulis(aplikacija). |
| Sužadinimo sąlyga: | Klientas nori daugiau informacijos apie detalę. |
| Po sąlyga: | Sugeneruojama ir atspausdinama etiketė. |
| Pagrindinis scenarijus: | 1. Sandėlininkas mobilaus telefono pagalba nuskaityto ant detalės esantį QR kodą; 2. Telefone pateikiama išsami informacija apie detalę; |
| Alternatyvūs scenarijai: | Nepavykus nuskaityti kodo ar detalės neradus DB parodomas klaidos pranešimas. |

4.6. FUNKCINIAI REIKALAVIMAI IR REIKALAVIMAI DUOMENIMS

4.6.1. Funkciniai reikalavimai

| | | |
|--------------------------------------|---|--------------------------------------|
| Reikalavimas: 1 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 1 |
| Aprašymas: | Administratorius arba sandėlininkas turi prisijungti prie sistemos, norint pradėti darbą | |
| Pagrindimas: | Kuomet administratorius arba sandėlininkas nėra prisijungęs prie sistemos jie negali ja naudotis. | |
| Šaltinis: | Įmonės darbuotojai | |
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog jis nėra prisijungęs | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|--------------------------------------|---|--------------------------------------|
| Reikalavimas: 2 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 2 |
| Aprašymas: | Administratorius prideda naują vartotoją į sistemą. | |
| Pagrindimas: | Įmonėje įsidarbina naujas darbuotojas. | |
| Šaltinis: | Administratorius | |
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog naujas vartotojas sėkmingai užregistruotas sistemoje. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|------------------------|--|--------------------------------------|
| Reikalavimas: 3 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 3 |
| Aprašymas: | Administratorius redaguoja esamo vartotojo duomenis. | |
| Pagrindimas: | Įmonėje sumažėja darbuotojų. | |
| Šaltinis: | Administratorius | |

| | |
|--------------------------------------|---|
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog vartotojo duomenys sėkmingai buvo atnaujinti. |
| Užsakovo tenkinimas: | 5 |
| <u>Užsakovo netenkinimas:</u> | 4 |
| <u>Konfliktai:</u> | Nėra |
| Priklausomybės: | Nėra |
| Papildoma medžiaga: | Nėra |
| Istorija: | 2016-03-08 |

| | | |
|--------------------------------------|---|--------------------------------------|
| Reikalavimas: 4 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 4 |
| Aprašymas: | Administratorius pašalina esamą vartotoją iš sistemos. | |
| Pagrindimas: | Įmonėje sumažėja darbuotojų. | |
| Šaltinis: | Administratorius | |
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog vartotojas sėkmingai pašalintas iš sistemos | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|--------------------------------------|--|--------------------------------------|
| Reikalavimas: 5 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 5 |
| Aprašymas: | Administratorius ar sandėlininkas prideda naują kategoriją. | |
| Pagrindimas: | Reikalinga sukurti kategorijų medį, kad automobilių detalės būtų logiškai sugrupuotos. | |
| Šaltinis: | Administratorius/Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog kategorija buvo sėkmingai sukurta | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |

| | |
|------------------|------------|
| Istorija: | 2016-03-08 |
|------------------|------------|

| | | |
|--------------------------------------|---|--------------------------------------|
| Reikalavimas: 6 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 6 |
| Aprašymas: | Administratorius ar sandėlininkas redaguoja kategorijos duomenis. | |
| Pagrindimas: | Reikalinga pakeisti kategorijos pavadinimas, statusą ar kitus parametrus. | |
| Šaltinis: | Administratorius/Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog kategorijos duomenys buvo sėkmingai atnaujinti. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|--------------------------------------|--|--------------------------------------|
| Reikalavimas: 7 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 7 |
| Aprašymas: | Administratorius ar sandėlininkas pašalina kategoriją iš sistemos. | |
| Pagrindimas: | Kategorija tampa nebeaktuali kataloge. | |
| Šaltinis: | Administratorius/Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog kategorijos duomenys buvo sėkmingai pašalinti iš sistemos. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|------------------------|--|--------------------------------------|
| Reikalavimas: 8 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 8 |
| Aprašymas: | Administratorius ar sandėlininkas prideda informaciją apie naują detalę. | |
| Pagrindimas: | Atsiranda detalių kurios nėra suvestos į sistemą. | |
| Šaltinis: | Administratorius/Sandėlininkas | |

| | |
|--------------------------------------|---|
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog detalė sėkmingai buvo išsaugota kataloge. |
| Užsakovo tenkinimas: | 5 |
| <u>Užsakovo netenkinimas:</u> | 4 |
| <u>Konfliktai:</u> | Nėra |
| Priklausomybės: | Nėra |
| Papildoma medžiaga: | Nėra |
| Istorija: | 2016-03-08 |

| | | |
|--------------------------------------|---|--------------------------------------|
| Reikalavimas: 9 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 9 |
| Aprašymas: | Administratorius ar sandėlininkas redaguoja esamos detalės duomenis. | |
| Pagrindimas: | Atsiranda poreikis pakeisti duomenis apie esamą detalę. | |
| Šaltinis: | Administratorius/Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog detalės duomenys buvo sėkmingai atnaujinti. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|--------------------------------------|--|---------------------------------------|
| Reikalavimas: 10 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 10 |
| Aprašymas: | Administratorius ar sandėlininkas pašalina detalę iš sistemos. | |
| Pagrindimas: | Detalė tampa nebeaktuali kataloge. | |
| Šaltinis: | Administratorius/Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema informuoja vartotoją, jog detalė buvo sėkmingai pašalinta iš sistemos. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |

| | |
|------------------|------------|
| Istorija: | 2016-03-08 |
|------------------|------------|

| | | |
|--------------------------------------|---|---------------------------------------|
| Reikalavimas: 11 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 11 |
| Aprašymas: | Administratorius ar sandėlininkas ieško informacijos apie detalę. | |
| Pagrindimas: | Klientas pateikia užklausą apie jį dominančią detalę. | |
| Šaltinis: | Administratorius/Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema pateikia paieškos rezultatus. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|--------------------------------------|--|---------------------------------------|
| Reikalavimas: 12 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 12 |
| Aprašymas: | Administratorius paržiūri sąrašą detalių kurios yra užsakytos. | |
| Pagrindimas: | Klientas užsako jam reikalingas detales. | |
| Šaltinis: | Administratorius | |
| Tinkamumo kriterijus: | Sistema pateikia sąrašą užsakytų detalių. | |
| Užsakovo tenkinimas: | 3 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|--------------------------------------|---|---------------------------------------|
| Reikalavimas: 13 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 13 |
| Aprašymas: | Sandėlininkas paržiūri sąrašą detalių kurios yra užsakytos nuardymui. | |
| Pagrindimas: | Klientas užsako jam reikalingas detales. | |
| Šaltinis: | Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema pateikia sąrašą užsakytų detalių nuardymui. | |
| Užsakovo tenkinimas: | 3 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |

| | |
|----------------------------|------------|
| <u>Konfliktai:</u> | Nėra |
| Priklausomybės: | Nėra |
| Papildoma medžiaga: | Nėra |
| Istorija: | 2016-03-08 |

| | | |
|--------------------------------------|---|---------------------------------------|
| Reikalavimas: 14 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 14 |
| Aprašymas: | Administratorius ar Sandėlininkas paržiūri sąrašą automobilių kurie šiuo metu yra ardomi. | |
| Pagrindimas: | Reikalinga sužinoti kokie automobiliai šiuo metu ardomi ir kokių detalių dar liko juose . | |
| Šaltinis: | Administratorius/Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema pateikia sąrašą automobilių kurie šiuo metu yra ardymo stadijoje ir rodo kokios detalės juose dar liko. | |
| Užsakovo tenkinimas: | 4 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|--------------------------------------|---|---------------------------------------|
| Reikalavimas: 15 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 15 |
| Aprašymas: | Sistema sugeneruoja sąskaitą faktūrą. | |
| Pagrindimas: | Klientui reikalinga išrašyti sąskaitą faktūrą. | |
| Šaltinis: | Administratorius | |
| Tinkamumo kriterijus: | Pateikiama sugeneruota ir paruošta spausdinimui sąskaita. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

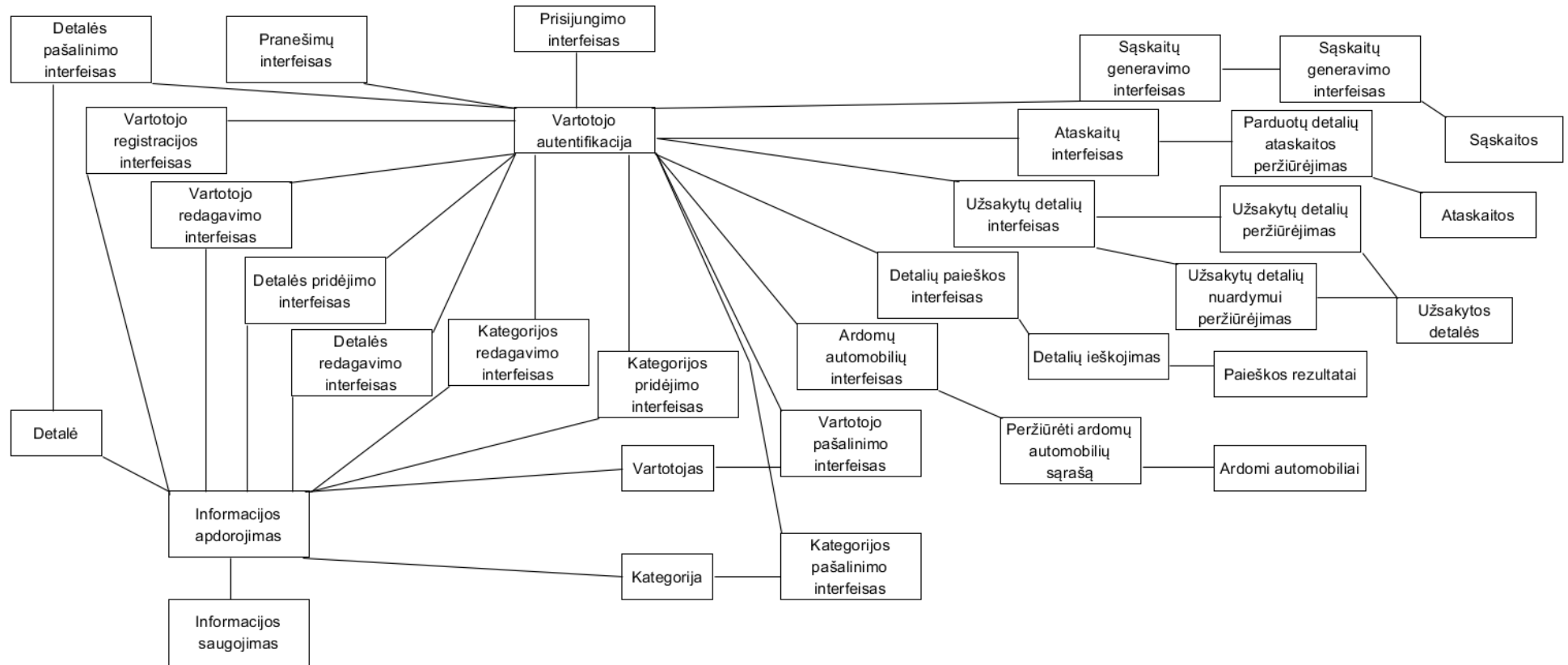
| | | |
|--------------------------------------|--|---------------------------------------|
| Reikalavimas: 16 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 16 |
| Aprašymas: | Sistema sugeneruoja ataskaitą apie parduotas detales. | |
| Pagrindimas: | Įmonės vadovas nori gauti informaciją apie parduotas dalis. | |
| Šaltinis: | Administratorius | |
| Tinkamumo kriterijus: | Sistema pateikia ataskaitą apie parduotas detales, pagal administratoriaus pasirinktus kriterijus. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|--------------------------------------|--|---------------------------------------|
| Reikalavimas: 17 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 17 |
| Aprašymas: | Sistema sugeneruoja etikete su QR kodu. | |
| Pagrindimas: | Etiketė reikalinga detalei identifikuoti. | |
| Šaltinis: | Sandėlininkas | |
| Tinkamumo kriterijus: | Sistema sugeneruoja etiketę su QR kodu ir leidžia ją išspausdinti. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

| | | |
|------------------------------|--|---------------------------------------|
| Reikalavimas: 18 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 18 |
| Aprašymas: | Sandėlininkas naudodamasis mobiliuoju telefonu kuriame yra įdiegta kuriamos sistemos aplikacija gali gauti detalią informaciją apie dominančią detalę. | |
| Pagrindimas: | Klientas prieš pirkdamas detalę nori įsitinkinti, kad ji tiks jo automobiliui. | |
| Šaltinis: | Sandėlininkas | |
| Tinkamumo kriterijus: | Nuskaičius QR kodą su mobiliuoju telefonu jame pateikiama detali informacija apie detalę. | |

| | | |
|--------------------------------------|-----------------------------|---------------------------------------|
| Reikalavimas: 18 | Reikalavimo tipas: F | Įvykis/panaudojimo atvejis: 18 |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-08 | |

4.6.2. Reikalavimai duomenims



10 pav. Klasių diagrama.

4.7. REIKALAVIMAI SISTEMOS IŠVAIZDAI

| | | |
|--------------------------------------|---|---|
| Reikalavimas: 19 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Sistemos sąsajai naudojami standartiniai saityno komponentai. | |
| Pagrindimas: | Sistema bus naudojama kasdien, todėl ja naudotis turi būti paprasta, langai, mygtukai, meniu ir kiti komponentai neturi išsiskirti. Tai leis darbuotojui lengviau priprasti prie naujos sistemos. | |
| Šaltinis: | Užsakovas | |
| Tinkamumo kriterijus: | Žinantis funkcinės sistemos galimybes įmonės darbuotojas sugebėtų naudotis sistema be apmokymo. | |
| Užsakovo tenkinimas: | 4 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-09 | |

4.8. REIKALAVIMAI PANAUDOJAMUMUI

| | | |
|--------------------------------------|---|---|
| Reikalavimas: 20 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Sistemos sąsajai naudojami standartiniai saityno komponentai. | |
| Pagrindimas: | Sistema bus naudojama kasdien, todėl ja naudotis turi būti paprasta, langai, mygtukai, meniu ir kiti komponentai neturi išsiskirti. Tai leis darbuotojui lengviau priprasti prie naujos sistemos. | |
| Šaltinis: | Užsakovas | |
| Tinkamumo kriterijus: | Žinantis funkcinės sistemos galimybes įmonės darbuotojas sugebėtų naudotis sistema be apmokymo. | |
| Užsakovo tenkinimas: | 4 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-09 | |

| | | |
|--------------------------------------|---|---|
| Reikalavimas: 21 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Įvykus klaidai sistema turi aiškiai apie tai pranešti. | |
| Pagrindimas: | Reikalinga, siekiant informuoti naudotoją apie neteisingą sistemos vartojimą. | |
| Šaltinis: | Vykdytojas | |
| Tinkamumo kriterijus: | Vedant neteisingus duomenis, sistema duomenų neišsaugo, bei praneša apie įvykusią klaidą. | |
| Užsakovo tenkinimas: | 4 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-09 | |

4.9. REIKALAVIMAI VYKDYMO CHARAKTERISTIKOMS

| | | |
|--------------------------------------|---|---|
| Reikalavimas: 22 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Sistema turi leisti vienu metu skaityti tuos pačius duomenis daugiau nei vienam vartotojui. | |
| Pagrindimas: | Sistema turės aptarnauti daugiau nei vieną vartotoją tuo pačiu metu. | |
| Šaltinis: | Vykdytojas | |
| Tinkamumo kriterijus: | Sistema leidžia matyti tuos pačius duomenis daugiau nei vienam vartotojui. | |
| Užsakovo tenkinimas: | 4 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-09 | |

4.10. REIKALAVIMAI VEIKIMO SĄLYGOMS

| | | |
|-------------------------|--|---|
| Reikalavimas: 23 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Sistema turi būti galima naudotis per interneto naršyklę. | |
| Pagrindimas: | Kompiuteryje nereikia diegti papildomos programinės įrangos. | |
| Šaltinis: | Vykdytojas | |

| | |
|--------------------------------------|--|
| Tinkamumo kriterijus: | Sistema gali naudotis bet kuris registruotas vartotojas, turintis prieigą prie kompiuterio su internetu ir interneto naršykle. |
| Užsakovo tenkinimas: | 5 |
| <u>Užsakovo netenkinimas:</u> | 5 |
| <u>Konfliktai:</u> | Nėra |
| Priklausomybės: | Nėra |
| Papildoma medžiaga: | Nėra |
| Istorija: | 2016-03-09 |

4.11. REIKALAVIMAI SISTEMOS PRIEŽIŪRAI

| | | |
|--------------------------------------|--|---|
| Reikalavimas: 23 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Sistemos darbuotojų klientinės darbo vietos neturi reikalauti pastovios sistemos administratoriaus priežiūros. | |
| Pagrindimas: | Naudingiau daryti sistemos pakeitimus vienoje vietoje, t.y. bazių ar aplikacijų serveriuose nei kiekvienoje klientinėje darbo vietoje. | |
| Šaltinis: | Vykdytojas | |
| Tinkamumo kriterijus: | Atliktus sistemos pakeitimus, kliento darbo vietų keisti nereikia. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-09 | |

| | | |
|--------------------------------------|--|---|
| Reikalavimas: 24 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Keičiantis organizacijos veiklos taisyklėms turi būti įmanoma adaptuoti sistemą prie naujų poreikių. | |
| Pagrindimas: | Nenaudinga kurti sistemą iš naujo, kai pasikeičia įmonės veiklos taisyklės. | |
| Šaltinis: | Vykdytojas | |
| Tinkamumo kriterijus: | Įvykus pokyčiams įmonėje, sistemos nereikia kurti iš naujo. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 4 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-09 | |

4.12. REIKALAVIMAI SAUGUMUI

| | | |
|--------------------------------------|---|---|
| Reikalavimas: 25 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Dirbti su sistemoje kaupiamais duomenimis gali tik sistemoje registruoti ir administratoriaus patvirtinti vartotojai. | |
| Pagrindimas: | Reikia užtikrinti kaupiamų duomenų saugumą. | |
| Šaltinis: | Užsakovas | |
| Tinkamumo kriterijus: | Neregistruoti vartotojai negali naudotis sistema. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-09 | |

| | | |
|--------------------------------------|--|---|
| Reikalavimas: 26 | Reikalavimo tipas: N | Įvykis/panaudojimo atvejis: Visi |
| Aprašymas: | Sistema žymisi kas kokias funkcijas vykdė. | |
| Pagrindimas: | Su sistema dirba daug vartotojų, kad nustatyti kas įvedė klaidingą informaciją. Taip pat atstatyti sugadintus duomenis, nelaimės atveju. | |
| Šaltinis: | Užsakovas | |
| Tinkamumo kriterijus: | Įvykdžius duomenų pakeitimą, prieš tai buvę duomenys yra išsaugomi. | |
| Užsakovo tenkinimas: | 5 | |
| <u>Užsakovo netenkinimas:</u> | 5 | |
| <u>Konfliktai:</u> | Nėra | |
| Priklausomybės: | Nėra | |
| Papildoma medžiaga: | Nėra | |
| Istorija: | 2016-03-09 | |

4.13. KULTŪRINIAI-POLITINIAI REIKALAVIMAI

Reikalavimų nėra

4.14. TEISINIAI REIKALAVIMAI

Reikalavimų nėra

4.15. NAUJOS PROBLEMOS

4.15.1. Problemos diegimo aplinkai

Problemu diegiant sistemą iškilti neturėtų kadangi naujai sukurta sistema veiks atskirame nutolusiame serveryje.

4.15.2. Įtaka jau instaliuotoms sistemoms

Nauja sistema neturės jokios įtakos įmonės jau naudojamai programinei įrangai, kadangi sistema veiks nutolusiame serveryje.

4.15.3. Neigiamas vartotojų nusiteikimas

Neigiama vartotojų reakcija gali kilti sistemos eksploatacijos pradžioje. Kadangi sistemos naudotojai nėra patyrę informacinių technologijų žinovai, todėl bus atlikti bendri apmokymai darbui su naująja sistema. Taip pat sistemos naudotojas, iškilus klausimui ar problemai, galės kreiptis pagalbos į sistemos priežiūrą atliekantį specialistą realiu laiku. Taip pat neigiamą vartotojų reakcija gali kilti dėl būtinybės suvesti susikaupusius duomenis (detales esančias sandėliuose). Šiuo atveju, priemonių, kurios galėtų padėti vartotojui, nėra numatyta. Stengiamasi, kad sistema funkcionuotų be jokių sutrikimų.

4.15.4. Kliudantys diegimo aplinkos apribojimai

Apribojimų nėra

4.15.5. Galimos naujos sistemos sukeltos problemos

Gali iškilti sistemos administravimo ar palaikymo problema, kol vartotojai pripras prie naujos sistemos.

4.16. UŽDAVINIAI

4.16.1. Sistemos pateikimo žingsniai (etapai)

- **Reikalavimų nustatymas:**

- poreikių surinkimas;
 - poreikių specifikavimas.

- **Analizė:**

- reikalavimų sistemai specifikavimas.

- **Projektavimas:**

- architektūros specifikavimas;
 - detalus architektūros specifikavimas.

- **Realizavimas:**

- funkcijų hierarchijos sudarymo modulis;
 - veiksmų, formuojančių rezultatus, diagramos modulis;

funkcinių modulių realizacija.

- **Testavimas:**
modulių testavimas;
bendras sistemos testavimas.
- **Diegimas ir eksploatavimas.**
- **Vartotojo apmokymas.**

4.16.2. Vystymo etapai

Pagrindiniai sistemos vystymo etapai:

Paraiška – projekto tikslų bei reikalavimų suderinimas tarp jo užsakovo, vadovo ir vykdytojų;

Darbų planas – projekto realizavimo darbų eigos, jų atlikimo terminų nustatymas;

Reikalavimų specifikacija – projekto pagrindinių apribojimų nustatymas, funkcinių ir nefunkcinių reikalavimų sudarymas, išeigos apskaičiavimas;

Detalios architektūros specifikacija – projekto detalios architektūros projektavimas;

Testavimo planas – projekto testavimo darbų eigos sudarymas, testavimo metodų ir kriterijų parinkimas;

Sistemos prototipas – realizuojamos sistemos supaprastinto varianto paruošimas;

Naudotojo dokumentacija – išsamus naudojimosi sistema aprašymas;

Programinės įrangos realizacija – sistemos programavimas bei pilnas ištestavimas;

Sistemos įdiegimas – programinės įrangos instaliavimas ir sistemos parengimas naudojimui.

4.17. PRITAIKYMAS

4.17.1. Reikalavimai esamų duomenų perkėlimui

Esami duomenys – įmonės sandėliuose esančių automobilių detalių informacija turės būti suvesti į naują sistemą rankiniu būdu.

4.17.2. Reikalingas duomenų transformavimas perkeliant į naują sistemą

Dėl esamų duomenų tipo duomenų transformacija negalima.

4.18. RIZIKOS

4.18.1. Galimos sistemos kūrimo rizikos

Galimos projekto rizikos ir jų tikimybės pateiktos sekančioje lentelėje.

28 lentelė. Sistemos kūrimo rizikos.

| Rizikos faktorius | Tikimybinis įvertinimas* |
|---|--------------------------|
| Reikalavimų specifikacijos pasikeitimai realizavimo fazėje | 10 |
| Besivystantys naudotojo reikalavimai | 9 |
| Kūrimo darbus atliekančio darbuotojo ligos atvejis ar būtina komandiruotė | 8 |
| Architektūros klaidų aptikimas realizavimo fazėje | 6 |

* – įvertinimo skalė: 10 – rizika tikrai sukels problemas, 1 – rizika nesukels problemų.

4.18.2. Atsitiktinumų (rizikų) planas

Rizikos sprendimų būdai pateikti sekančioje lentelėje.

29 lentelė. Atsitiktinumų (rizikų) planas).

| Rizikos faktorius | Problemos sprendimas |
|---|---|
| Reikalavimų specifikacijos pasikeitimai realizavimo fazėje | Sistema kuriama pagal užsakovo ir vykdytojo nustatytą specifikaciją, kurios keitimas iki sistemos galutinio pridavimo įmanomas tik sutarus abiem pusėm. Nesant šiam sutarimui, pakeitimai atliekami sistemos vystymo etape. |
| Besivystantys naudotojo reikalavimai | Bus pateikiamas sistemos prototipas |
| Kūrimo darbus atliekančio darbuotojo ligos atvejis ar būtina komandiruotė | Sistemos kūrimo terminai turės būti pratęsti. |

4.19. VARTOTOJO DOKUMENTACIJA IR APMOKYMAS

Kadangi sistemos vartotojai nebus patyrę informacinių technologijų žinovai ir darbui su sistema reikės naujų darbo kompiuteriu įgūdžių, todėl bus pateikiama vartotojo dokumentacijos:

- Vartotojo vadovas (dokumentas);
 - Naujo automobilio įvedimas į sistemą
 - Naujos detalės įvedimas į sistemą
 - Paieška sistemoje
 - Užsakymo – detalių nuardymui formavimas
 - Veiksmai su detale
 - Kategorijų medžio valdymas (detalių medžio sudarymas)
 - Darbuotojų valdymas
 - Ataskaitos
 - Filtrų taikymas

Dokumentaciją rengia sistemos kūrėjas, vartotojai dokumentacijos rengime nedalyvauja.

Dokumentacija pateikiama Microsoft Word failų pavidale.

Numatoma atlikti ir vartotojų apmokymus kurių trukmė iš viso bus 8 val. Apmokymų metu bus supažindinama su bendrais darbo su sistema principais.

4.20. PERSPEKTYVINIAI REIKALAVIMAI

Projekto sėkmės atveju tikimasi pritaikyti produktą rinkos sąlygoms, todėl projektuojant turi būti atsižvelgta į projekto išplečiamumą.

4.21. IŠVADOS

1. Skyriuje pateikti reikalavimai yra specifikuoti pasinaudojant Volere šablonu, ir padeda suprasti bendrą sistemos funkcionalumą.

5. ARCHITEKTŪROS SPECIFIKACIJA

5.1. ARCHITEKTŪROS PATEIKIMAS

Sistemos architektūra bus pateikta remiantis RUP, t.y. modelį sudarys 5 vaizdai:

- Panaudojimo;
- Loginis;
- Procesų;
- Komponentų išdėstymo;
- Duomenų.

Visos diagramos bus vaizduojamos UML notacija:

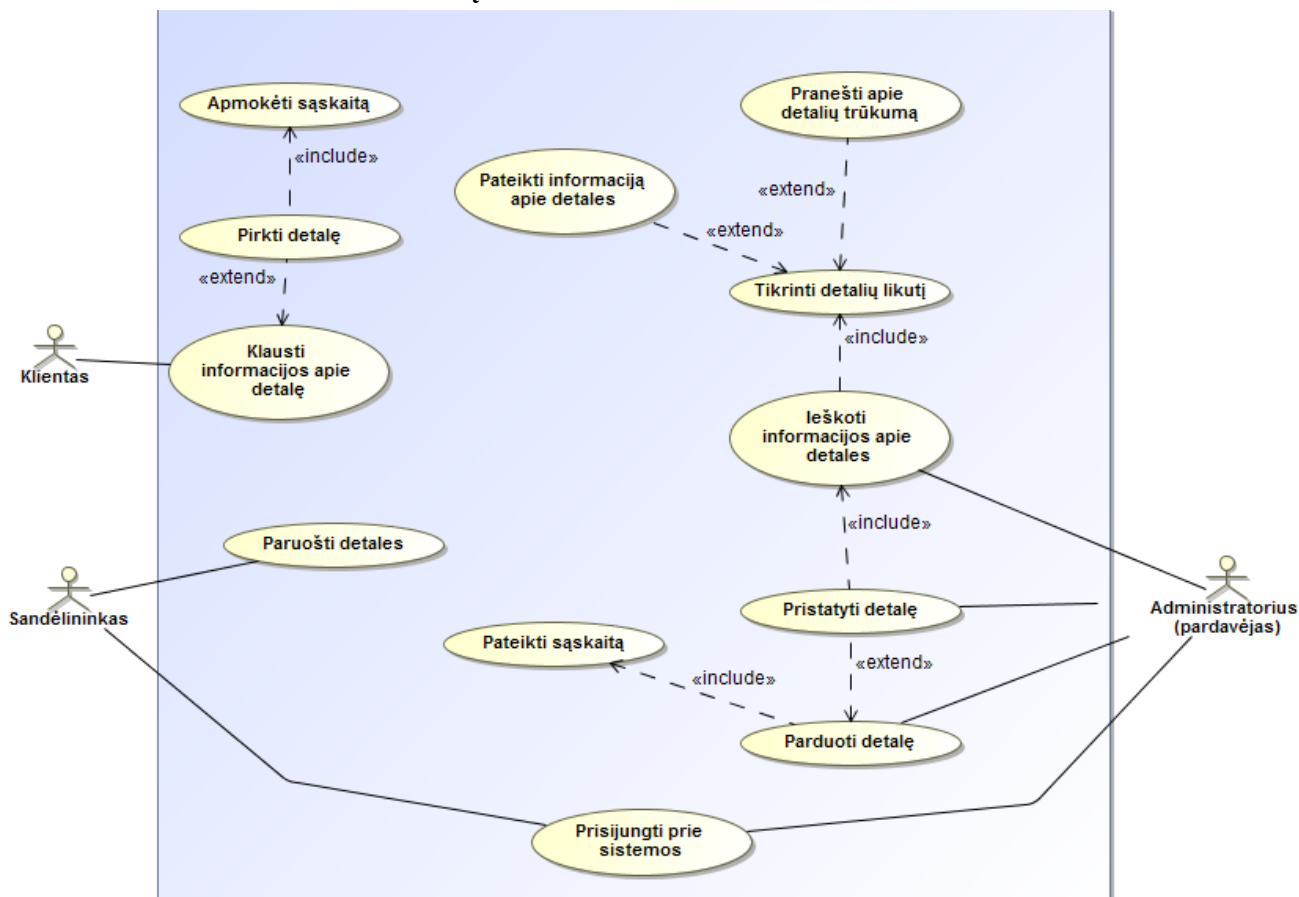
- Panaudojimo vaizdas – panaudojimo atvejų diagrama;
- Loginis – sistemos išskaidymo i paketus diagrama;
- Procesų – bendradarbiavimo ir būsenų diagramos;
- Komponentų išdėstymo – išdėstymo diagrama;
- Duomenų – duomenų struktūros diagrama.

5.2. ARCHITEKTŪROS TIKSLAI IR APRIBOJIMAI

Sistema turi būti suprojektuota taip, kad ja būtų galima lengvai išplėsti ar prijungi naujus modulius. Sistema turi neleisti neautorizuotiems vartotojams prie jos prisijungti. Sudarant sistemos architektūrą, turi būti atsižvelgta i reikalavimų specifikacijoje apibrėžtas programos vykdymo charakteristikas.

Projektui realizuoti bus naudojama internetinių puslapių kūrimo programavimo kalba PHP (Laravel 5) susieta su duomenų bazių paketu MySQL ir skriptų programavimo kalba JavaScript. Be to šie paketai yra nemokami ir nereikalauja papildomo įdiegimo lėšų įmonių kompiuteriuose. Pakanka to, kad kompiuteriai turėtų prieigą prie interneto. Diagramos vaizduojamos UML kalba ir kaip modeliavimo įrankis yra pasirinkta „MagicDraw“

5.3. PANAUDOJIMO ATVEJŲ VAIZDAS



11 pav. Esminiai panaudojimo atvejai.

5.4. SISTEMOS STATINIS VAIZDAS

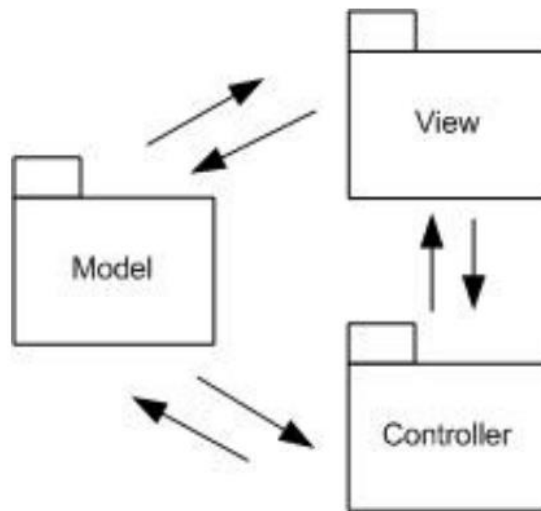
5.4.1. Apžvalga

Sistema bus kuriama MVC (Model-View-Controller) principu. MVC veikimas yra paremtas separation of concerns (SoC) principu. Tai reiškia, kad programa yra išskaidoma į tokias dalis, kad jų funkcionalumas/poreikiai nepersidengintų. Pačiame MVC šis principas pasireiškia logikos, prezentacijos ir turinio atskyrimu. Trys dalys atsakingos už savo „veiklos sritį“.

Model – tam tikros srities duomenų valdymo/pateikimo vienetas – realizuojamas pagrindinis programos algoritmas;

View – suteikia programos modeliui išvaizdą, tinkamą vartotojui suprasti (formos, sąrašai) – realizuojama grafinė vartotojo sąsaja;

Controller – atsako už modelio valdymą bei „vaizdų“ pateikimus pagal atitinkamus veiksmus (vartotojo interakciją): mygtuko paspaudimas, teksto įvedimas ir panašiai – atliekamas duomenų perdavimas nuo View modulio Modeliui ir atgal.



12 pav. MVC architektūros veikimas.

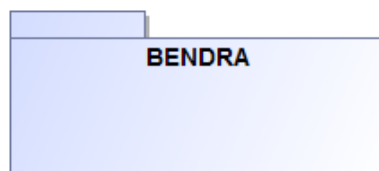
Skyriuje aprašoma sistemos loginė struktūra. Pateikiamas sistemos išskaidymas į paketus. Vadovaujantis MVC architektūros principu, visas projektas dalinamas į tris pagrindinius modulius:

Vartotojo sąsaja – įvedimo ir išvedimo formos, per kurias vartotojas sąveikauja su sistema;

Vartotojo administravimas – administravimo sąsajos realizavimas;

Duomenų mainai – duomenų su kuriais dirba vartotojas saugojimo ir perdavimo elementai.

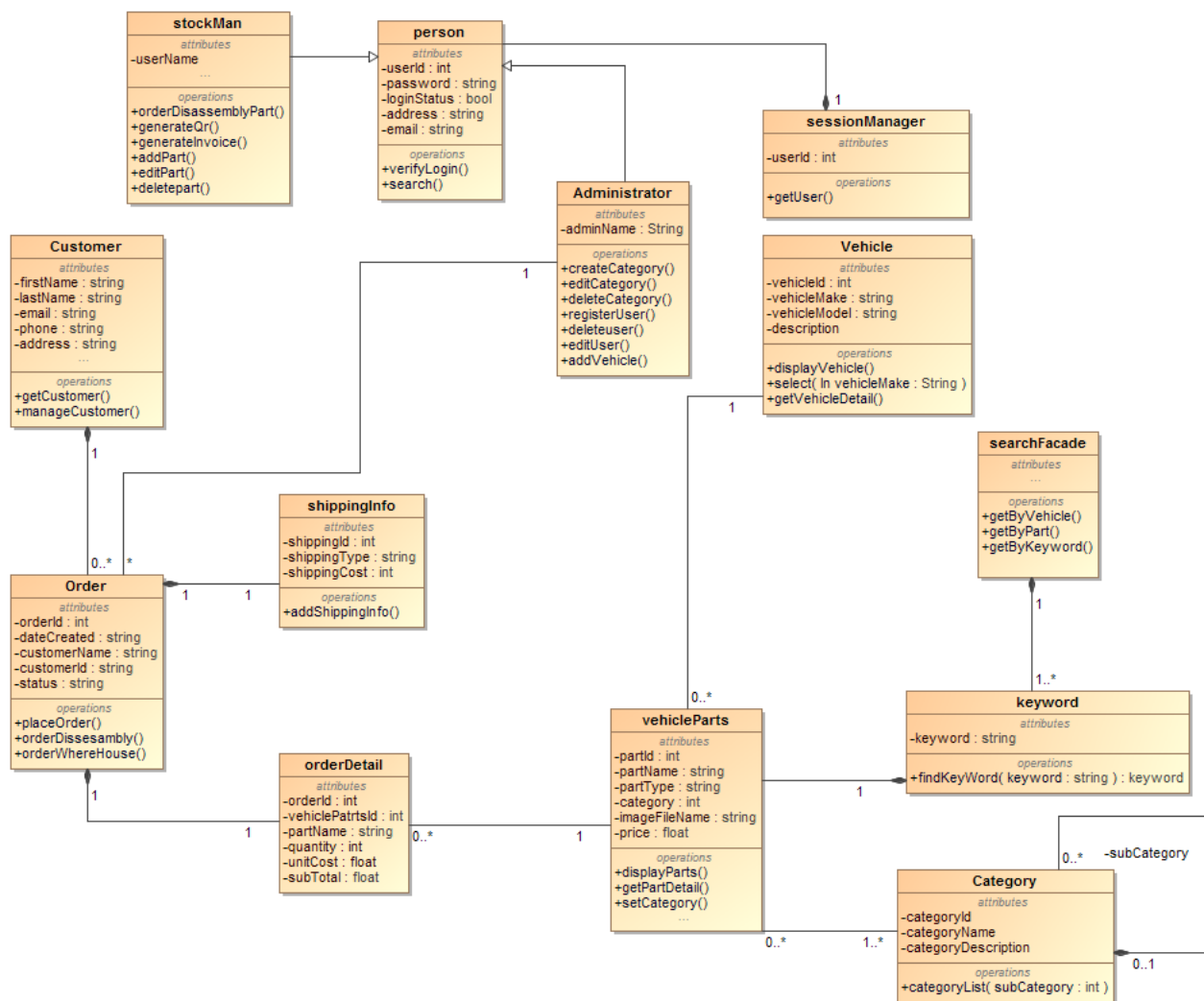
Sistema formaliai sudaryta iš vieno paketo ir gali būti atvaizduota, kaip parodyta:



13 pav. Kuriamos sistemos paketas.

5.4.2. Paketų detalizavimas

Sistemos (pateikto paketo) suskaidymas į klases pateiktas paveiksle žemiau:



14 pav. Klasių diagrama.

Person – klasė skirta valdyti sistemos naudotojų veiksmus. Jos pagrindinis metodas `verifylogin()`, kviečiamas kada vartotojas sistemoje paspaudžia mygtuką prisijungti. Metodas grąžina `true` ir sukuriama sesija jei įvesti prisijungimo duomenys buvo teisingi, kitu atveju `false` su klaidos pranešimu jeigu įvesti prisijungimo duomenys buvo klaidingi.

Administrator – klasė valdo sistemos administratoriaus veiksmus, taip pat ši klasė paveldi klasės `person` metodus. **Klasė turi metodus:**

createCategory() – skirtas kurti kategorijas, kategorijų „gylis“ nėra ribojamas.

editCategory() – skirtas pakeisti sukurtos kategorijos duomenis ar vietą kategorijų medyje.

deleteCategory() – skirtas ištrinti nepageidaujamą kategoriją.

registerUser() – skirtas sukurti naują sistemos naudotoją.

editUser() – skirtas redaguoti esamo sistemos naudotojo duomenis.

deleteUser() – skirtas pašalinti sistemos naudotoją.

addVehicle() – skirtas pridėti naują automobilį į sistemą.

stockMan – klasė skirta valdyti sandėlininko veiksmus kaip ir Administrator klasė ji taip pat paveldi klasės Person metodus.

Klasė turi metodus:

orderDisassemblyPart() – skirtas peržiūrėti detalių sąrašą kurias reikia nuardyti nuo automobilių.

generateQR() – skirtas qr kodo sugeneravimui ir jo atspausdinimui.

generateInvoice() – skirtas sąskaitos faktūros generavimui ir spausdinimui.

addPart() – skirtas automobilio detalės pridėjimui į katalogą.

editPart() – skirtas detalės esančios kataloge duomenims redaguoti.

deletePart() – skirtas detalės pašalinimui iš katalogo.

sessionManager – klasė palaiko sistemos naudotojui reikalingas operacijas. **getUser()** metodas kviečiamas kada reikalinga gauti informaciją apie prisijungusį naudotoją.

vehicle – klasė skirta valdyti sistemoje esančius automobilių duomenis.

Klasė turi metodus:

displayVehicle() – skirtas atvaizduoti sistemoje esančius automobilius.

select() – skirtas pasirinkti, kad automobiliai būtų rodomi pagal automobilio gamintoją ir modelį.

getVehicleDetail() – skirtas atvaizduoti automobilio detalią informaciją.

vehicleParts – klasė atsakinga už automobilio detalių pateikimą pagal kategoriją.

Klasė turi metodus:

displayParts() – skirtas pateikti sistemoje esančių detalių sąrašą.

getPartDetail() – skirtas pateikti informaciją apie konkrečią pasirinktą detalę.

setCategory() – skirtas nustatyti detalės kategoriją.

Order – klasė skirta valdyti klientų užsakymams.

Klasė turi metodus:

placeOrder() – skirtas užsakymų patvirtinimui.

orderDisassembly() – skirtas detalių kurios nėra nuardytos nuo automobilio užsakymams.

orderWhereHouse() – skirtas detalių rezervacijai.

Customer – klasė skirta valdyti su klientais susijusią informaciją

getCustomer() – skirtas gauti klientų sąrašą ir atlikti paiešką jame.

manageCustomer() – skirtas detalios informacijos peržiūrai apie klientą, jo duomenų redagavimui, ar pašalinimui iš sistemos.

searchFacade – klasė atsakinga už informacijos paiešką sistemoje.

Klasė turi metodus:

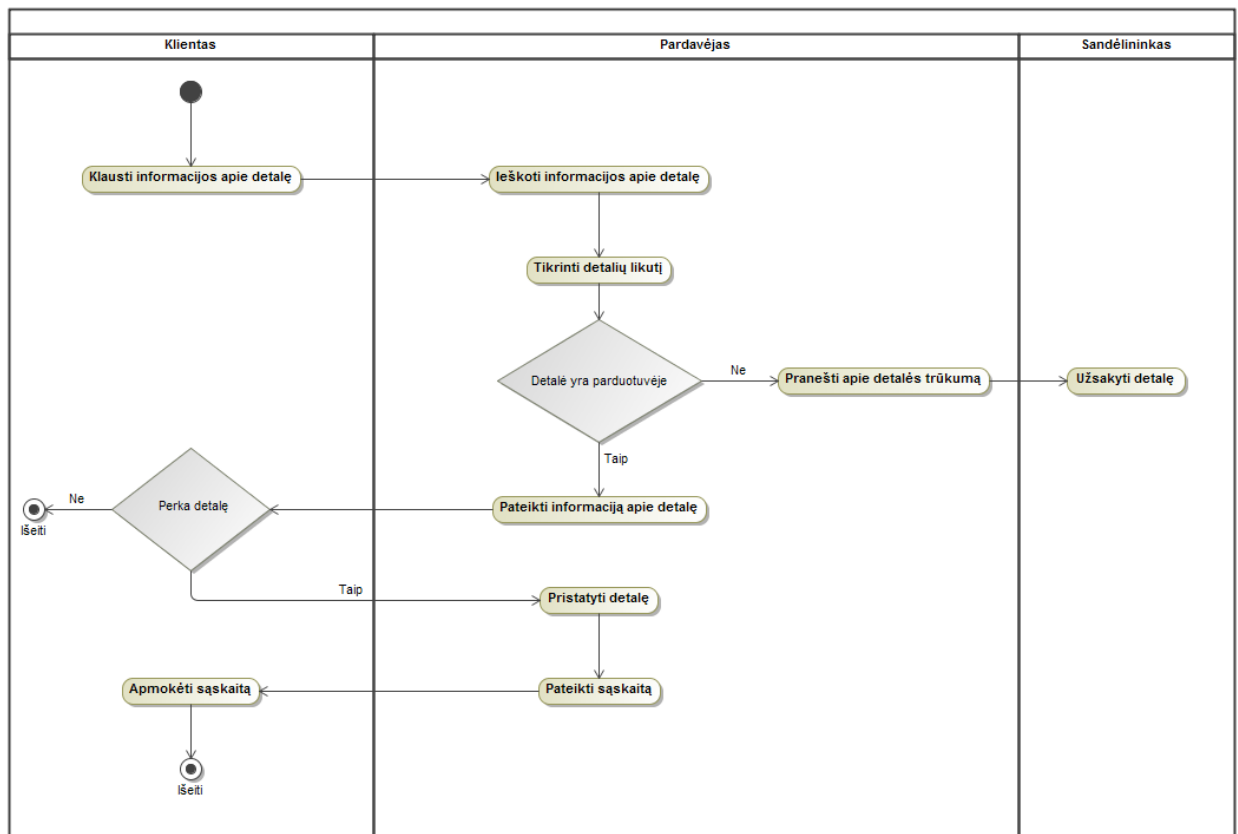
getByVehicle() – skirtas atlikti paiešką pagal automobilio modelį.

getByPart() – skirtas atlikti paiešką pagal automobilio detalę.

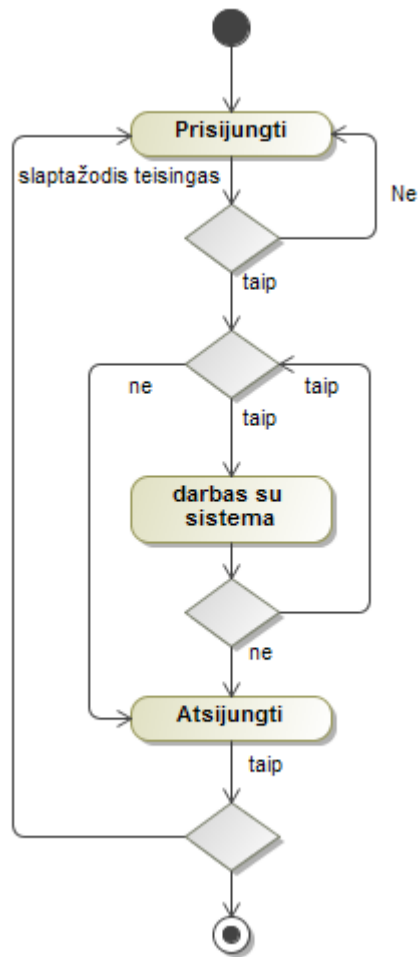
getByKeyword() – skirtas rasti informaciją pagal detalės raktažodžius.

5.5. SISTEMOS DINAMINIS VAIZDAS

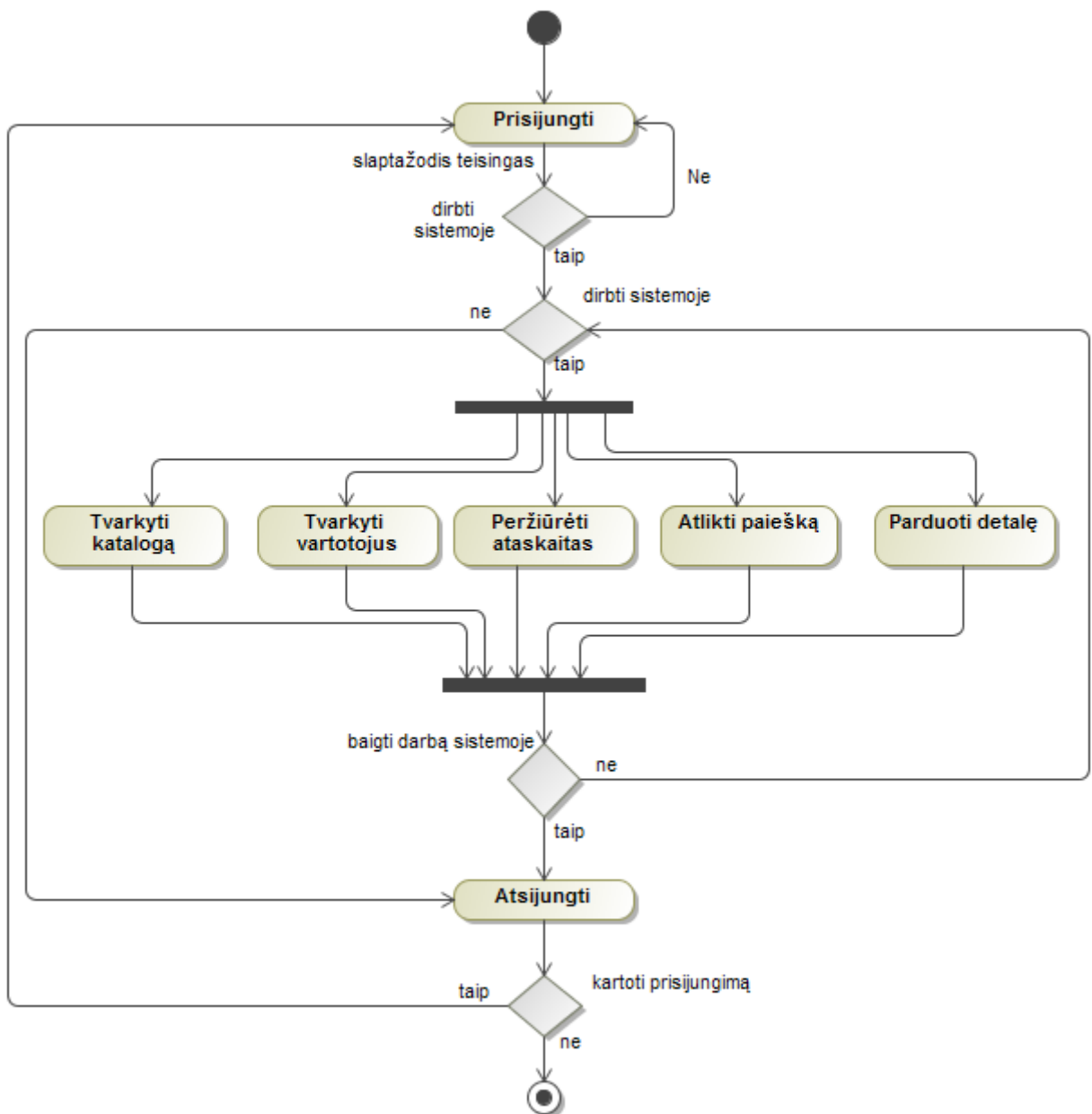
Šiame skyrelyje pateikiami sistemos objektų būsenų, sistemos elementų bendradarbiavimo diagramų pavyzdžiai.



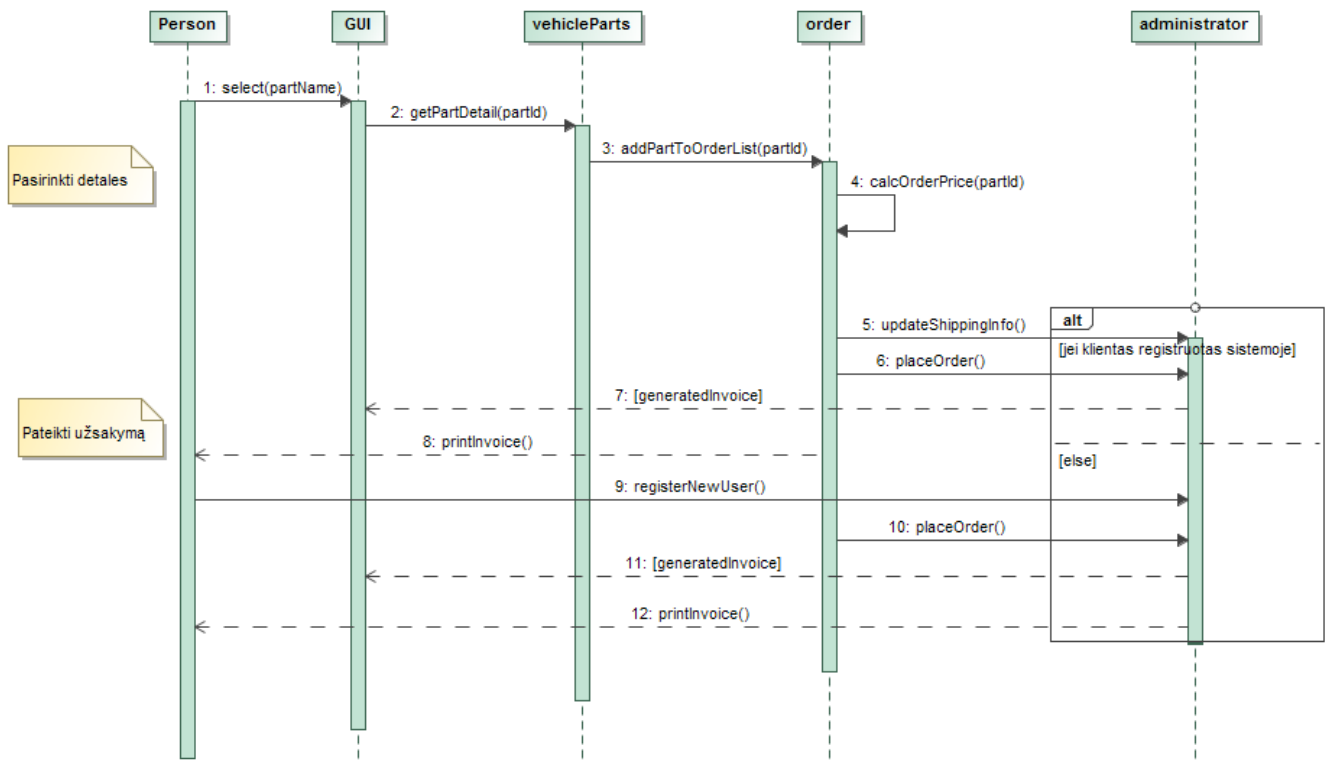
15 pav. Bendroji verslo schema.



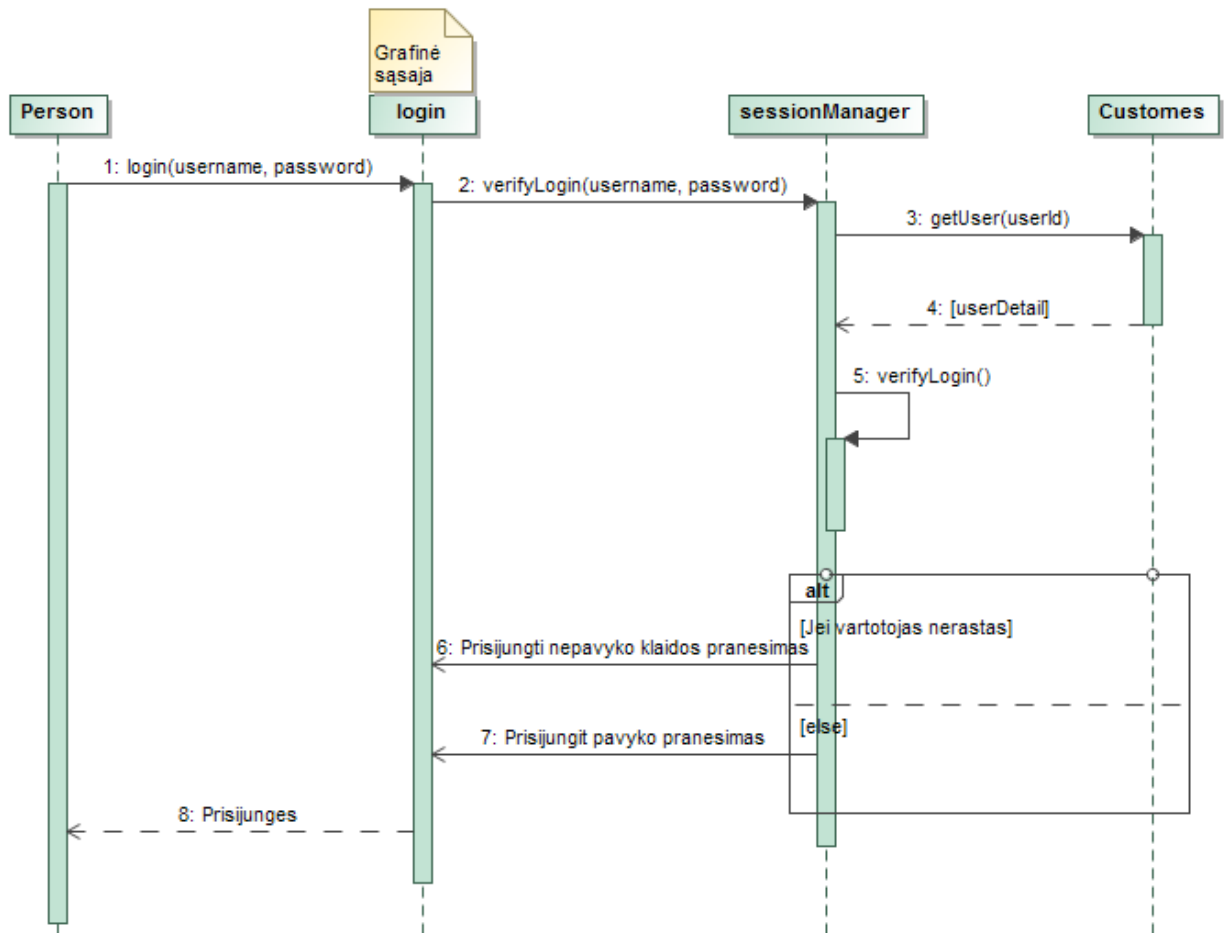
16 pav. Sandėlininko prisijungimo prie sistemos veiklos diagrama.



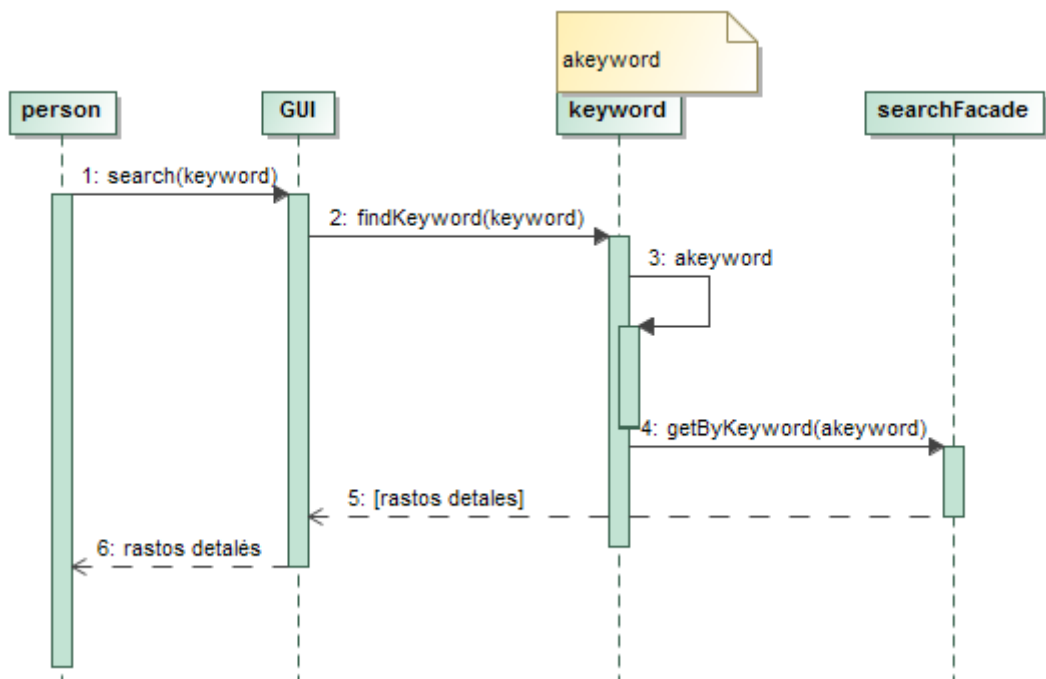
17 pav. Sistemos administratoriaus prisijungimo ir darbo sistemoje veiklos diagrama.



18 pav. Detalių užsakymo procesas, sekų diagrama.

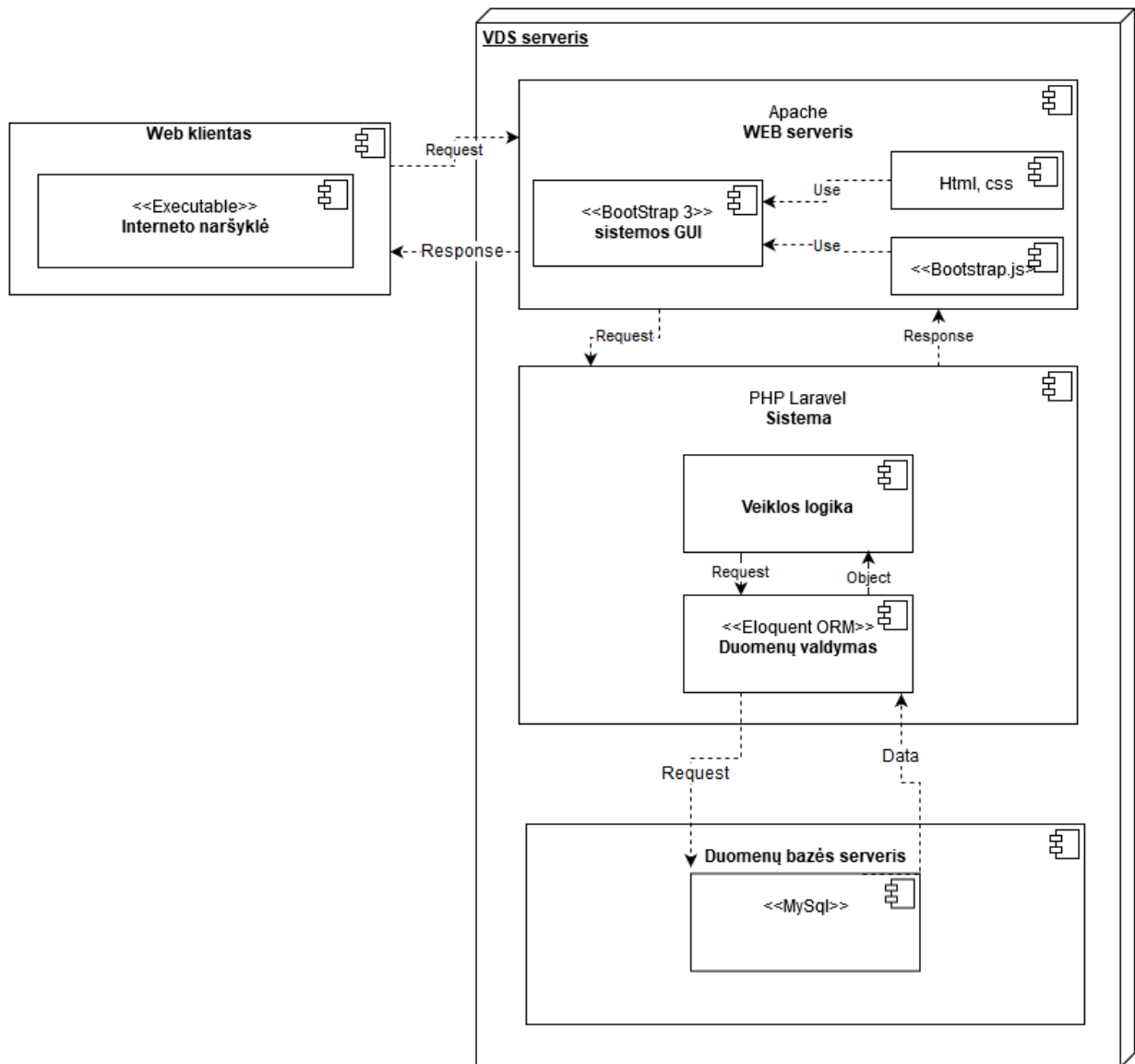


19 pav. Naudotojų prisijungimas prie sistemos, sekų diagrama.



20 pav. Informācijas paieška sistemoje, sekų diagrama.

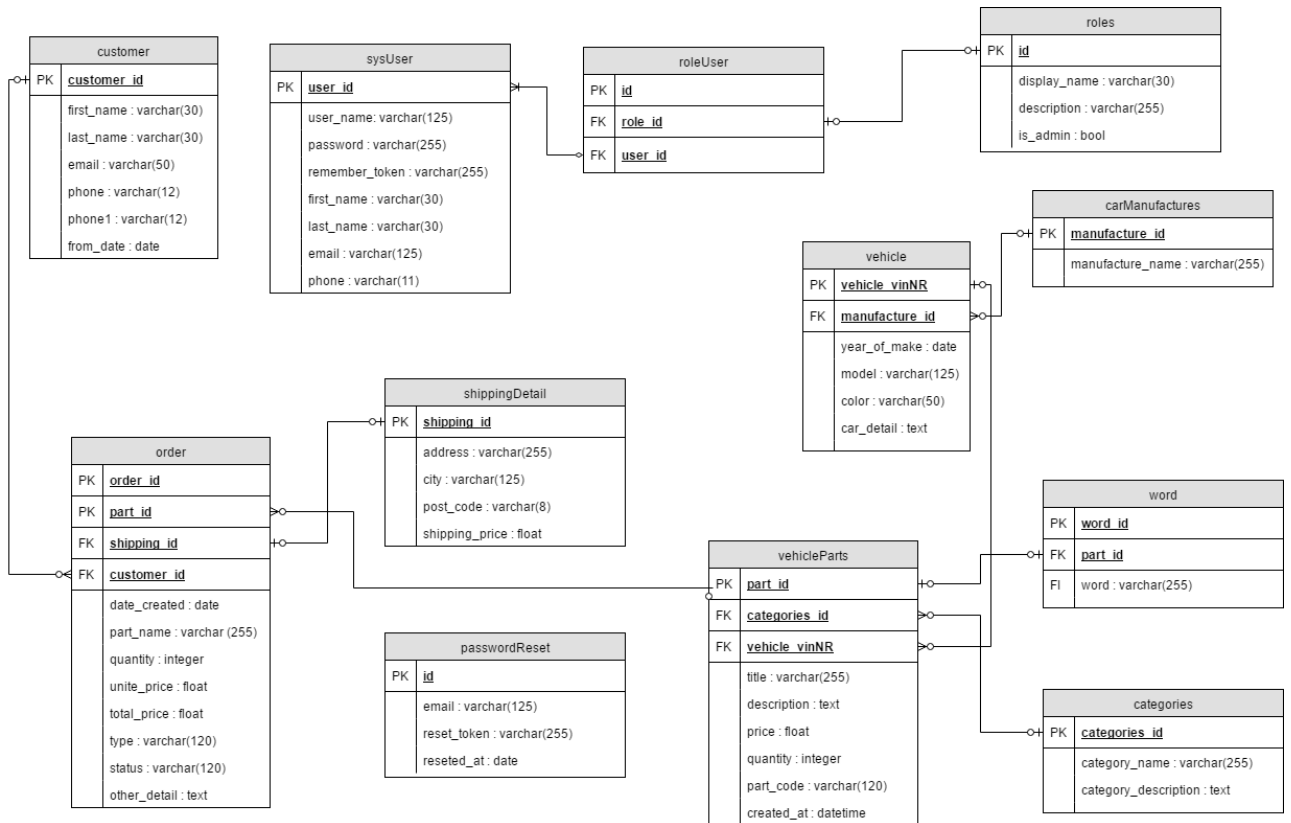
5.6. IŠDĖSTYMO VAIZDAS



21 pav. Sistemos išsidėstymo vaizdas.

Visa sistema bus patalpinta Linux OS operacine sistema valdomame serveryje. Taip pat sistemai naudojamas „Apache“ tinklo serveris. WEB aplikacijos veiklos logikoje talpinami projektuojamos sistemos algoritmai, sisteminiai procesai, skaičiavimo logika. ORM – kuris priskiriamas prie duomenų valdymo, programuojamas pasitelkiant „Laravel Eloquent“ karkasą. Duomenų valdiklis paimtus duomenis iš duomenų šaltinio perduoda objektams sukurti ar duomenims saugoti. Duomenų saugykla, tai sistemos reliacinė duomenų bazė, kuri veikia MYSQL 5 versijos serveryje.

5.7. DUOMENŲ VAIZDAS



22 pav. Duomenų bazės vaizdas.

DB komponentas skirtas visų sistemoje naudojamų duomenų saugojimui ir tvarkymui. DB komponento pagrindinė funkcija – duomenų saugojimas ir duomenų integralumo užtikrinimas. Duomenys saugomi lentelėse. Duomenų integralumas užtikrinamas raktiniais laukais, ryšiais tarp lentelių ir trigeriais.

5.8. IŠVADOS

1. Pasirinktas MVC architektūrinis modelis ir Laravel 5 programavimo karkasas kartu su MySQL5 duomenų bazių valdymo sistema leis nesunkiai realizuoti sistemos pakeitimus, visi pakeitimai bus vykdomi serveriuose nekeičiant klientinių darbo vietų.
2. Sistemos architektūros specifikacijos leis lengviau praplėsti ar pakeisti sistemos funkcionalumą pagal naujai iškilusius poreikius, be to taip užtikrinamos reikalavimų specifikacijoje (4 skyriuje) minimos sistemos savybės.

6. TESTAVIMO MEDŽIAGA

6.1. TESTAVIMO PLANAS

Šiame skyriuje aprašomos testavimo procedūros ir strategija, testuojama programinė įranga kartu su savo sąsajomis. Numatoma, kad testavimo strategija apims šias veiklas: vienetų testavimą, integravimo testavimą, priėmimo testavimą ir aukšto lygio testavimą. Taip apibrėžiami laukiami testavimo rezultatai, turimi testavimo resursai.

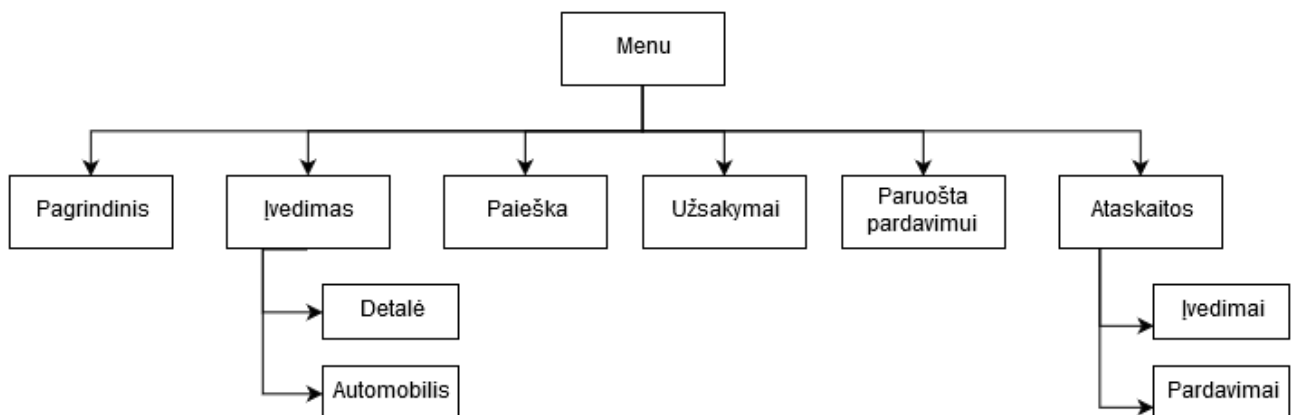
Aprašomi testavimo metodai, kurie bus naudojami testuojant programinę įrangą. Visi metodai aprašomi abstrakčiai, nepateikiamos detalios metodų realizavimo procedūros, jos bus pateiktos sekančiame skyriuje.

Skyriuje taip pat aprašoma testavimo procedūra žvelgiant iš testavimo procedūros perspektyvos.

6.1.1. Testuojama programinė įranga

Vartotojo sąsajos meniu hierarchija

2 pav. pateikta naudotų automobilio dalių sandėlio valdymo sistemos meniu hierarchijos struktūra.



23 pav. Meniu hierarchijos schema.

Žemiau esančiuose skyriuose pateikiami kiekvieno meniu punkto ir jo atliekamo veiksmo aprašymai.

„Pagrindinis“ meniu punktas

Paspaudus šį meniu punktą rodomas pagrindinis sistemos langas su pagrindine sistemos informacija (kiek detalių yra sistemoje, naujausios detalės ir automobiliai sistemoje, darbuotojai, detalių pardavimo statistika einamąjį laikotarpį ir kt.) .

„Įvedimas -> Detalė“ meniu punktas

Paspaudus šį meniu punktą atidaromas programos langas su forma skirta įvesti naują detalę į sistemą.

„Įvedimas -> Automobilis“ meniu punktas

Paspaudus šį meniu punktą atidaromas programos langas su forma skirta įvesti naują detalę į sistemą.

„Paieška“ meniu punktas

Paspaudus šį meniu punktą atidaromas langas padedantis greitai surasti reikiamą detalę sistemoje. Paieška vyksta iš detalių medžio pasirinkus reikiamą kategoriją ir suvedus automobilio parametrus (markė, modelis, modifikacija), taip pat galima ir tekstinė paieška.

„Užsakymai“ meniu punktas

Paspaudus šį meniu punktą atidaromas langas skirtas valdyti užsakymams kurie yra priimami iš klientų – reikiamos klientui detalės neradus sandėlyje galima atlikti šios detalės užsakymą tai yra nuardyti nuo aikštelėje esamų automobilių.

„Paruošta pardavimui“ meniu punktas

Paspaudus šį meniu punktą atidaromas langas kuriame pateikiamos visos sandėlyje esančios automobilio detalės. Yra galimybė detales filtruoti pagal automobilį, įvedimo datą, darbuotoją kuris įvedė detalę į sistemą.

„Ataskaitos -> Įvedimai“ meniu punktas

Paspaudus šį meniu punktą atidaromas langas kuriame pateikiama detalių įvedimo ataskaita (kiek detalių ir kurio darbuotojo detalės buvo įvestos į sistemą per pasirinktą laikotarpį).

„Paruošta pardavimui“ meniu punktas

Paspaudus šį meniu punktą atidaromas langas kuriame pateikiama pardavimo ataskaita per pasirinktą laikotarpį.

6.1.2. Vienetų testavimas

Vienetų testavimo etape bus tikrinama ar pagal metodui perduotus pradinius duomenis gaunami laukiami rezultatai. Duomenys parenkami neatsižvelgiant į metodo struktūrą „juoda dėžė“. Testiniai atvejai bus sudaromi pagal sistemos specifikaciją.

Vienetų testavimas tai pats pirmas testavimo etapas, kurį atlieką patys programuotojai. Jie turi išsiaiškinti ar modulis veikia tinkamai. Atsiradusias klaidas turi pašalinti nedelsiant. Vėlesniame etape kartu su integravimo testavimu vienetų testus turi atlikti ir atskiras testuotojas.

6.1.3. Priėmimo testavimas

Testavimas atliekamas pagal „juodos dėžės“ principą užbaigtam produktui. Sistema nuodugnai ištestuojama pagal reikalavimų specifikaciją ir patikrinama ar sistema atitinką vartotojo poreikius. Radus neatitikimą tarp sistemos ir vartotojo poreikių, tai patikrinama reikalavimų specifikacijoje. Jei specifikacija atitinka vartotojo poreikius, reiškia sistema neatitinka specifikacijos ir bus registruojama klaida. Jei specifikacijoje nerastas vartotojo poreikis, registruojamas reikalingas sistemos patobulinimas, kuris įgyvendinamas per pokyčių valdymą sekančioje sistemos versijoje.

Priėmimo testavimas bus atliekamas dviem etapais: be sistemos vartotojo ir su sistemos vartotoju. Be sistemos vartotojo bus atliekamas nuodugnus vartotojo sąsajos testavimas patikrinant kiekvieną jos elementą bei sistemos funkcijos pagal specifikaciją.

6.1.4. Aukšto lygio testavimas

Užbaigto, integruoto produkto galutinis testavimas. Bus patikrinta kaip sistema geba atlaikyti stresinio testavimo metu gautas užduotis, taip pat bus atliktas saugumo testavimas naudojant „Acunetix Vulnerability“ skenerį kuris aptinka įvairaus tipo internetinius pažeidžiamumus.

6.2. TESTAVIMO RESURSAI

Techninė ir programinė įranga: jokių papildomų techninių ar programinių resursų, kurie viršytų jau turimus resursus skirtus sistemos kūrimui, nereikia. Sistemos testavimas atliekamas ant darbinės DB serverio, klientinė dalis atitinka standartinės kompiuterizuotos darbo vietos parametrus.

6.3. TESTAVIMO ĮRANKIAI IR APLINKA

Programinės įrangos testavimui bus naudojama „PHPStorm 10 IDE“ programavimo aplinka ir „PHPUni“t testavimo karkasas. Saugumo testavimui bus naudojamas „Acunetix web vulnerability“ skeneris.

6.4. TESTAVIMO PROCEDŪROS

Skyriuje pateikiamos testavimo procedūros, kurios bus naudojamos atliekant programinės įrangos testavimą.

vienetų testavimas: žemiau lentelėse pateikiami atskirų vienetų testavimo atvejai.

30 lentelė. Prisijungimo lango testavimo atvejai.

| Testas | Laukiamas rezultatas |
|--|--|
| Įvedami teisingi prisijungimo duomenys | Parodomas pagrindinis programos langas su naudotojo lygį atitinkančiais meniu punktais |
| Įvedami neteisingi prisijungimo duomenys | Naudotojas informuojamas, kad įvesti klaidingi duomenys |

31 lentelė. Pagrindinio meniu testavimo atvejai.

| Testas | Laukiamas rezultatas |
|--|---|
| Pasirenkamas meniu punktas Įvedimas->detalė | Atidaromas langas kuriame pateikiama forma skirta įvesti naują detalę į sistemą. |
| Pasirenkamas meniu punktas Įvedimas->automobilis | Atidaromas langas kuriame pateikiama forma skirta įvesti naują automobilį į sistemą. |
| Pasirenkamas meniu punktas Paieška | Atidaromas langas kuriame pateikiamas detalių ieškojimo vedlys. |
| Pasirenkamas meniu punktas Užsakymai | Atidaromas langas kuriame pateikiama informacija apie šiuo metu užsakytas detales. |
| Pasirenkamas meniu punktas Paruošta pardavimui | Atidaromas langas kuriame pateikiama informacija apie šiuo metu sandėlyje turimas detalių atsargas. |
| Pasirenkamas meniu punktas Ataskaitos ->įvedimai | Atidaromas langas kuriame pateikiama detalių ir automobilių įvedimo ataskaita į sistemą. |
| Pasirenkamas meniu punktas Ataskaitos ->pardavimai | Atidaromas langas kuriame pateikiamos detalių pardavimo ataskaitos. |

32 lentelė. Naujos detalės įvedimo į sistemą testavimo atvejai.

| Testas | Laukiamas rezultatas |
|--|--|
| Teisingais duomenimis užpildoma detalės įvedimo forma ir paspaudžiamas mygtukas Išsaugoti | Gaunamas pranešimas, kad detalė sėkmingai pridėta į sistemą. |
| Detalės įvedimo forma užpildoma nelogiškais duomenimis (skaičiai vietoj raidžių ir atvirkščiai) ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas apie neteisingai įvestus duomenis. |
| Detalės įvedimo forma užpildoma nepilnai (privalomi laukai pažymėti *) neužpildomi, ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas apie būtinus užpildyti laukus. |

33 lentelė. Naujo automobilio įvedimo į sistemą testavimo atvejai.

| Testas | Laukiamas rezultatas |
|--|---|
| Teisingais duomenimis užpildoma automobilio įvedimo forma ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas, kad automobilis sėkmingai pridėta į sistemą. |
| Automobilio įvedimo forma užpildoma nelogiškais duomenimis (skaičiai vietoj raidžių ir atvirkščiai) ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas apie neteisingai įvestus duomenis. |
| Automobilio įvedimo forma užpildoma nepilnai (privalomi laukai pažymėti *) neužpildomi ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas apie būtinus užpildyti laukus. |

34 lentelė. Detalių paieška sistemoje testavimo atvejai.

| Testas | Laukiamas rezultatas |
|---|---|
| Iš kategorijų medžio pasirenkamos kategorijos kuriose bus atliekama paieška, pasirenkama informacija apie automobilį (markė, metai, modifikacija), įvedamas paieškos tekstas, paspaudžiamas mygtukas ieškoti. | Gaunami paieškos rezultatai pagal pasirinktą kategoriją, automobilį ir įvestą paieškos tekstą |
| Iš kategorijų medžio pasirenkamos kategorijos kuriose bus atliekama paieška, pasirenkama informacija apie automobilį (markė, metai, modifikacija), paspaudžiamas mygtukas ieškoti. | Gaunami paieškos rezultatai pagal pasirinktą kategoriją, automobilį. |
| Pasirenkamas visas kategorijų medis ir automobilis (markė, metai, modifikacija) ir paspaudžiamas mygtukas ieškoti. | Gaunami rezultatai su visomis detalėmis esančiomis sandėlyje pagal pasirinktą automobilį. |
| Iš kategorijų medžio pasirenkamos kategorijos kuriose bus atliekama paieška ir paspaudžiamas mygtukas ieškoti. | Gaunamas klaidos pranešimas, kad būtina pasirinkti automobilį ir jo markę. |
| Pasirenkamas norimas automobilis ir paspaudžiamas mygtukas ieškoti. | Gaunamas klaidos pranešimas, kad būtina pasirinkti kategoriją kurioje bus atliekama paieška. |

35 lentelė. Užsakymų valdymo testavimo atvejai.

| Testas | Laukiamas rezultatas |
|---|---|
| Iš užsakymo sąrašo pasirenkamas atsitiktinis užsakymas ir paspaudžiamas mygtukas peržiūrėti užsakovą. | Gaunama detali užsakymo informacija. |
| Atsidariusiame užsakymo lange paspaudžiamas mygtukas atšaukti užsakymą. | Gaunamas pranešimas, kad užsakymas sėkmingai atšauktas ir užsakymo informacija perkelta į atšauktų užsakymų sąrašą. |
| Atsidariusiame užsakymo lange paspaudžiamas mygtukas užsakymas įvykdytas | Gaunamas pranešimas, kad užsakymas sėkmingai įvykdytas ir užsakymas priskiriamas įvykdytų užsakymų grupei. |

36 lentelė. Pardavimų valdymo testavimo atvejai.

| Testas | Laukiamas rezultatas |
|---|--|
| Iš detalių sąrašo pasirenkama atsitiktinė detalė ir paspaudžiamas mygtukas detali informacija. | Gaunama detali informacija (įvedimo data ir kuris darbuotojas atliko įvedimą į sistemą, automobilio duomenys nuo kurio detalė buvo nuardyta, detalės kaina, QR kodas, nuotraukos). |
| Atsidariusiame detaliame detalės peržiūros režime paspaudžiamas mygtukas parduoti. | Gaunamas detalės pardavimo langas. |
| Detalės pardavimo lange pasirenkama nuolaida detalei ir paspaudžiamas mygtukas parduoti. | Gaunama pranešimas apie detalės kainą su pritaikyta nuolaida ir informacija, kad detalė parduota. |
| Iš detalių sąrašo pasirenkamos kelios detalės ir paspaudžiamas mygtukas generuoti sąskaitą ir parduoti. | Gaunama sugeneruota sąskaita faktūra. |
| Sąskaitos faktūros generavimo lange pasirenkama nuolaida ir paspaudžiamas mygtukas parduoti. | Išspausdinama sąskaita faktūra, gaunamas pranešimas apie sąskaitos sumą su nuolaida ir informacija, kad detalės sėkmingai parduotos. |

37 lentelė. Ataskaitų valdymo testavimo atvejai.

| Testas | Laukiamas rezultatas |
|--|---|
| Atsidariusiame lange Ataskaitos -> įvedimai pasirenkama data nuo iki ir paspaudžiamas mygtukas suformuoti. | Gaunama detali detalių įvedimo ataskaita per pasirinktą laikotarpį. |
| Atsidariusiame lange Ataskaitos -> įvedimai iš sąrašo darbuotojai pasirenkamas darbuotojas ir paspaudžiamas mygtukas suformuoti. | Gaunama detali darbuotojo įvestų detalių ataskaita. |
| Atsidariusiame lange Ataskaitos -> pardavimai pasirenkamas laikotarpis nuo iki | Gaunama detalių pardavimo ataskaita per pasirinktą laikotarpį. |

6.4.1. Priėmimo testavimas

Šis testavimas bus atliekamas kaip juodos dėžės testas. Jo pagrindinė užduotis patikrinti, kaip naudotų automobilio dalių sandėlio valdymo sistema atitinka reikalavimų specifikaciją.

Testavimo metu bus tikrinama ar pateikiant duomenis jie yra tinkamai apdorojami ir gražinama reikalingi rezultatai.

6.4.2. Aukšto lygio testavimas

Šiame testavime bus atliekamas programinės įrangos našumo testavimas simuliuojant didelius detalių kiekius sistemoje, taip pat patikrintas saugumas naudojantis „Acunetix web vulnerability“ skeneriu.

6.4.3. Testavimo resursų paskirstymas

Personalas

Programuotojas/testuotojas testuos visą sukurtą sistemą bendrai, bei jos atskirus modulius ir funkcionalumą.

6.5. TESTAVIMO REZULTATAI

6.5.1. Rankinio testavimo duomenys ir rezultatai

Žemiau lentelėse pateikiami atskirų vienetų atlikti testavimo atvejai ir rezultatai.

38 lentelė. Prisijungimo lango testavimo atvejai.

| Testas | Laukiamas rezultatas | Rezultatas |
|--|---|------------|
| Įvedami teisingi prisijungimo duomenys | Parodomas pagrindinis programos langas sunaudotojo lygį atitinkančiais meniu punktais | Pavyko |
| Įvedami neteisingi prisijungimo duomenys | Naudotojas informuojamas, kad įvesti klaidingi duomenys | Pavyko |

39 lentelė. Pagrindinio meniu testavimo atvejai.

| Testas | Laukiamas rezultatas | Rezultatas |
|--|--|------------|
| Pasirenkamas meniu punktas Įvedimas->detalė | Atidaromas langas kuriame pateikiama forma skirta įvesti naują detalę į sistemą. | Pavyko |
| Pasirenkamas meniu punktas Įvedimas->automobilis | Atidaromas langas kuriame pateikiama forma skirta įvesti naują automobilį į sistemą. | Pavyko |
| Pasirenkamas meniu punktas Paieška | Atidaromas langas kuriame pateikiamas detalių ieškojimo vedlys. | Pavyko |
| Pasirenkamas meniu punktas Užsakymai | Atidaromas langas kuriame pateikiama informacija apie šiuo metu užsakytas detales. | Pavyko |

| | | |
|---|---|--------|
| Pasirenkamas meniu punktas Paruošta pardavimui | Atidaromas langas kuriame pateikiama informacija apie šiuo metu sandėlyje turimas detalių atsargas. | Pavyko |
| Pasirenkamas meniu punktas Ataskaitos - >įvedimai | Atidaromas langas kuriame pateikiama detalių ir automobilių įvedimo ataskaita į sistemą. | Pavyko |
| Pasirenkamas meniu punktas Ataskaitos - >pardavimai | Atidaromas langas kuriame pateikiamos detalių pardavimo ataskaitos. | Pavyko |

40 lentelė. Naujos detalės įvedimo į sistemą testavimo atvejai.

| Testas | Laukiamas rezultatas | Rezultatas |
|--|--|------------|
| Teisingais duomenimis užpildoma detalės įvedimo forma ir paspaudžiamas mygtukas Išsaugoti | Gaunamas pranešimas, kad detalė sėkmingai pridėta į sistemą. | Pavyko |
| Detalės įvedimo forma užpildoma nelogiškais duomenimis (skaičiai vietoj raidžių ir atvirkščiai) ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas apie neteisingai įvestus duomenis. | Pavyko |
| Detalės įvedimo forma užpildoma nepilnai (privalomi laukai pažymėti *) neužpildomi, ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas apie būtinus užpildyti laukus. | Pavyko |

41 lentelė. Naujo automobilio įvedimo į sistemą testavimo atvejai.

| Testas | Laukiamas rezultatas | Rezultatas |
|--|---|------------|
| Teisingais duomenimis užpildoma automobilio įvedimo forma ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas, kad automobilis sėkmingai pridėta į sistemą. | Pavyko |
| Automobilio įvedimo forma užpildoma nelogiškais duomenimis (skaičiai vietoj raidžių ir atvirkščiai) ir | Gaunamas pranešimas apie neteisingai įvestus duomenis. | Pavyko |

| | | |
|---|--|--------|
| paspaudžiamas mygtukas Išsaugoti. | | |
| Automobilio įvedimo forma užpildoma nepilnai (privalomi laukai pažymėti *) neužpildomi ir paspaudžiamas mygtukas Išsaugoti. | Gaunamas pranešimas apie būtinus užpildyti laukus. | Pavyko |

42 lentelė. Detalių paieška sistemoje testavimo atvejai.

| Testas | Laukiamas rezultatas | Rezultatas |
|---|---|------------|
| Iš kategorijų medžio pasirenkamos kategorijos kuriose bus atliekama paieška, pasirenkama informacija apie automobilį (markė, metai, modifikacija), įvedamas paieškos tekstas, paspaudžiamas mygtukas ieškoti. | Gaunami paieškos rezultatai pagal pasirinktą kategoriją, automobilį ir įvestą paieškos tekstą | Pavyko |
| Iš kategorijų medžio pasirenkamos kategorijos kuriose bus atliekama paieška, pasirenkama informacija apie automobilį (markė, metai, modifikacija), paspaudžiamas mygtukas ieškoti. | Gaunami paieškos rezultatai pagal pasirinktą kategoriją, automobilį. | Pavyko |
| Pasirenkamas visas kategorijų medis ir automobilis (markė, metai, modifikacija) ir paspaudžiamas mygtukas ieškoti. | Gaunami rezultatai su visomis detalėmis esančiomis sandėlyje pagal pasirinktą automobilį. | Pavyko |
| Iš kategorijų medžio pasirenkamos kategorijos kuriose bus atliekama paieška ir paspaudžiamas mygtukas ieškoti. | Gaunamas klaidos pranešimas, kad būtina pasirinkti automobilį ir jo markę. | Pavyko |
| Pasirenkamas norimas automobilis ir paspaudžiamas mygtukas ieškoti. | Gaunamas klaidos pranešimas, kad būtina pasirinkti kategoriją kurioje bus atliekama paieška. | Pavyko |

43 lentelė. Užsakymų valdymo testavimo atvejai.

| Testas | Laukiamas rezultatas | Rezultatas |
|---|---|------------|
| Iš užsakymo sąrašo pasirenkamas atsitiktinis užsakymas ir paspaudžiamas mygtukas peržiūrėti užsakovą. | Gaunama detali užsakymo informacija. | Pavyko |
| Atsidariusiame užsakymo lange paspaudžiamas mygtukas atšaukti užsakymą. | Gaunamas pranešimas, kad užsakymas sėkmingai atšauktas ir užsakymo informacija perkelta į atšauktų užsakymų sąrašą. | Pavyko |
| Atsidariusiame užsakymo lange paspaudžiamas mygtukas užsakymas įvykdytas | Gaunamas pranešimas, kad užsakymas sėkmingai įvykdytas ir užsakymas priskiriamas įvykdytų užsakymų grupei. | Pavyko |

44 lentelė. Pardavimų valdymo testavimo atvejai.

| Testas | Laukiamas rezultatas | Rezultatas |
|---|--|------------|
| Iš detalių sąrašo pasirenkama atsitiktinė detalė ir paspaudžiamas mygtukas detali informacija. | Gaunama detali informacija (įvedimo data ir kuris darbuotojas atliko įvedimą į sistemą, automobilio duomenys nuo kurio detalė buvo nuardyta, detalės kaina, QR kodas, nuotraukos). | Pavyko |
| Atsidariusiame detaliame detalės peržiūros režime paspaudžiamas mygtukas parduoti. | Gaunamas detalės pardavimo langas. | Pavyko |
| Detalės pardavimo lange pasirenkama nuolaida detalei ir paspaudžiamas mygtukas parduoti. | Gaunama pranešimas apie detalės kainą su pritaikyta nuolaida ir informacija, kad detalė parduota. | Pavyko |
| Iš detalių sąrašo pasirenkamos kelios detalės ir paspaudžiamas mygtukas generuoti sąskaitą ir parduoti. | Gaunama sugeneruota sąskaita faktūra. | Pavyko |
| Sąskaitos faktūros generavimo lange pasirenkama nuolaida ir paspaudžiamas mygtukas parduoti. | Išspausdinama sąskaita faktūra, gaunamas pranešimas apie sąskaitos sumą su nuolaida ir informacija, kad detalės sėkmingai parduotos. | Pavyko |

45 lentelė. Ataskaitų valdymo testavimo atvejai.

| Testas | Laukiamas rezultatas | Rezultatas |
|--|---|------------|
| Atsidariusiame lange Ataskaitos -> įvedimai pasirenkama data nuo iki ir paspaudžiamas mygtukas suformuoti. | Gaunama detali detalių įvedimo ataskaita per pasirinktą laikotarpį. | Pavyko |
| Atsidariusiame lange Ataskaitos -> įvedimai iš sąrašo darbuotojai pasirenkamas darbuotojas ir paspaudžiamas mygtukas suformuoti. | Gaunama detali darbuotojo įvestų detalių ataskaita. | Pavyko |
| Atsidariusiame lange Ataskaitos -> pardavimai pasirenkamas laikotarpis nuo iki | Gaunama detalių pardavimo ataskaita per pasirinktą laikotarpį. | Pavyko |

6.6. IŠVADOS

1. Vienetų sistemos testavimas parodė, kad visos sistemos funkcijos veikia taip, kaip turėtų ir nesukeltų problemų besinaudojantiems sistema. Prisijungimas veikia tinkamai ir nukreipia naudotojus į jiems privilegijos teisėmis suteiktus resursus. Automobilio detalių įvedimas į sistemą veikia sklandžiai ir duomenų neiškraipo, paieška sistemoje veikia tinkamai ir pateikia rezultatus pagal pasirinktus paieškos parametrus.
2. Atlikus sistemos skenavimą su saugumo skeneriu „Acunetix web vulnerability“, rezultatai parodė, kad sistema yra apsaugota nuo kritinių klaidų ir veikia be sutrikimų. Sistema apsaugota nuo SQL atakų, nes visi sistemos laukai užklausoje paduodami kaip parametrai, todėl papildomos komandos tuose vietose neveikia, nėra tiesioginio įterpimo parametrų į užklausas. Taip pat sistemai negresia XSS atakų galimybė, nes visas tekstas puslapiuose spausdinamas su žyme „<h:outputText value="" />“, dėl ko tekstas atspausdinamas toks koks įrašytas su visomis „<script>“ žymėmis jei tokios buvo, taigi meta simboliai galimi naudoti naudotojų tiesiog kaip žymėjimas.

7. ŽODŽIŲ PAIEŠKA SQL PAGRINDO RELIACINĖSE DUOMENŲ BAZĖSE (TYRIMAS)

7.1. ĮVADAS

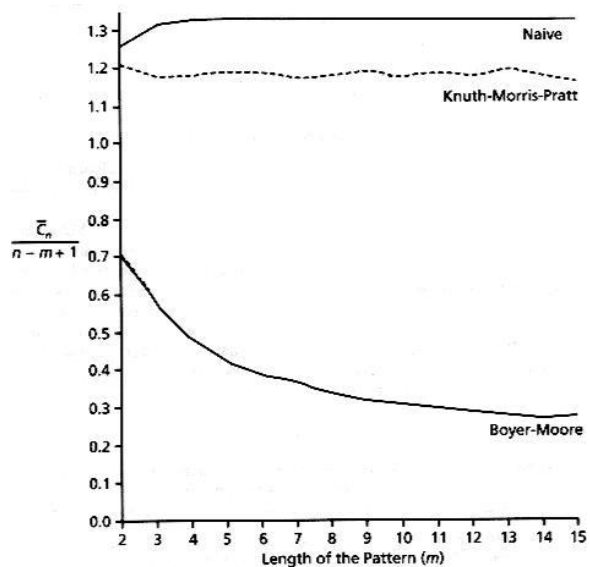
Šio skyriaus tikslas yra ištirti skirtingų teksto paieškos algoritmų efektyvumą, naudojant SQL pagrindo reliacinę duomenų bazę. Šio tyrimo rezultatai gali suteikti vertingos informacijos duomenų bazių projektavimui ir įvairių sistemų kūrimui kuriose naudojami simbolių eilutės tipo paieškos algoritmai.

Norint rasti tinkamą žodį žodyne reikia peržvelgti žodyną ir surasti žodį. Tai reikalauja tam tikrų išteklių, kad palyginti visus žodžius iš žodyno su žodžio paieškos šablonu. Yra nemažai skirtingų algoritmų, skirti simbolių eilutės atitikimo problemai spręsti. Kaip teigia R. Rivest [16] norint surasti m ilgio šabloną n ilgio tekste (kur $n > m$) paieškos laikas yra $O(n)$ blogiausiu atveju (kada m reikšmė fiksuota). Be to, blogiausiu atveju, reikia patikrinti mažiausiai $n - m + 1$ ženklų. Simbolių eilutės paieškos sudėtingumas gali skirtis priklausomai nuo naudojamo algoritmo.

Pats žinomiausias simbolių eilutės tipo paieškos algoritmas yra brutaliųjų jėgų algoritmas (ang. brute force). Jis bando palyginti bet kurią m ilgio sub-eilutę tekste su šablonu [17] todėl reikia patikrinti $n - m + 1$ skirtingas vieno žodžio kombinacijas, kad rasti visus šablono atitikimus. Šis algoritmas palygina visas šablono raides visoms žodžio kombinacijoms. Tuo tarpu Knuth–Morris–Pratt algoritmas [18] optimizuoja ir netikrina visų šablonų. Šis algoritmas lygindamas dvi eilutes, palygina iki pirmų raidžių neatitikimo, todėl algoritmo sudėtingumas mažesnis ir paieškos laikas labiau priklausomas nuo paieškos šablono ir lyginamos eilutės.

Siekdami sumažinti paieškos algoritmo sudėtingumą Boyer ir Moore [19] pasiūlė ieškoti šablono iš dešinės į kairę. Neįvykus neatitikimui šablonas randamas. Kitaip šablono lyginimas pastumiamas į kairę raidžių skaičiumi, kuris apskaičiuojamas naudojant euristinį algoritmą (reguliuojantį pagal atitikimo šablono skirtumą).

Eilučių paieškos algoritmo priklausomai nuo šablono ilgio tyrimas buvo atliktas Ricardo A. Baeza-Yates [20] (žr. Pav. 24).



24 pav. Eilutės paieškos algoritmų sudėtingumo palyginimas [5]

Jis parodė, Boyer ir Moore algoritmo trūkumus, lyginant su brutalią jėgą ir Knuth–Morris–Pratt algoritmais. Tačiau visi šie klasikiniai algoritmai buvo pasiūlyti 1977 m., tuo tarpu naujos informacinės technologijos siūlo įvairius problemos sprendimo būdus.

7.2. ŽODŽIŲ PAIEŠKA DUOMENŲ BAZĖJE

Reliacinės duomenų bazės modelis yra duomenų organizavimas į dvi matas lenteles vadinamas relacijomis [21]. Šio tipo duomenų valdymo sistemos dažniausiai yra skirtingose informacinių sistemų rūšyse [21, 22]. Reliacinių duomenų bazių valdymų sistemų populiarumas padidėjo dėka SQL, kuris leido sintaksei ištraukti reikiamus duomenis iš reliacinių duomenų bazių. Struktūrinė užklausų kalba (angl. Structured Query Language (SQL)) yra standartizuota kalba naudojama gauti ir atnaujinti duomenis, saugomus reliacinėse duomenų bazių lentelėse [20]. SQL turi skirtingų užklausų ir parametrų tipus. Tačiau eilučių ieškojimo algoritmai dažniausiai naudoja LIKE sakinį. **Jis turi du pakaitos simbolius:**

(%) – reiškia nulį, vieną ar kelis simbolius;

(_) reiškia vienintelį skaitmenį ar simbolį [21].

Simboliai gali būti panaudojami junginiuose, sudarančiuose norimą paieškos eilutės šabloną. Tačiau pats svarbiausias teksto paieškos elementas yra sistemos duomenų bazės projektavimas: duomenų bazė turi saugoti norimą rasti tekstą, o žodžio paieškos sistema dažniausiai ieško atitinkančių žodžių pagal tam tikrą šabloną, todėl naudojama daug prisijungimų prie duomenų bazės.

Norint pagreitinti LIKE sakinių vykdymą galima naudoti indeksus, kalbant suprantamai, indeksai tai yra tokia speciali duomenų struktūra, kurioje galima greitai atlikti paiešką. Kalbant apie indeksus MySQL, bendru atveju viskas susiveda į dvejetainio medžio (angl. Binary tree) struktūrą. Dvejetainių duomenų medžių yra įvairių, todėl svarbu prisiminti, kad MySQL naudoja surikiuotus dvejetainius medžius arba dvejetainius paieškos medžius (angl. binary search/ordered tree). Paprastai sutinkamas „btree“ trumpinys.

FULLTEXT indeksas – MySQL duomenų bazių valdymo sistema nuo 4.0 versijos turi galimybę indeksuoti DB lentelėse saugomą tekstinį turinį. Tam naudojamas FULLTEXT indeksavimas. Indeksuoti galima CHAR, VARCHAR ir TEXT tipų laukus.

Žemiau yra pateikiami penki duomenų bazių projektavimo pavyzdžiai.

Paprasčiausias žodžio paieškos duomenų bazės projektavimas parodytas pav. 25



25 pav. Duomenų bazės projektavimas Nr. 1.

Ši architektūra yra labai paprasta ir leidžia turėti kelis skirtingus žodynus (kelias DB lenteles kuriose bus ieškoma) toje pačioje sistemoje. Naudojant šį duomenų bazių projektavimą, gali būti naudojamos dviejų tipų SQL užklausos, skirtos gauti atitinkančius žodžius iš duomenų bazės:

- Atrinkti visus atitinkančius žodžius iš duomenų bazės pagal šabloną PTRN. Pvz: *SELECT word.word FROM word WHERE word.word LIKE „PTRN“*.
- Atrinkti visus atitinkančius žodžius iš tam tikros lentelės vehicleParts duomenų bazėje pagal šabloną PTRN. Pvz: *SELECT word.word FROM word vehicleParts WHERE word.word_id = vehicleParts.part_id AND word.word LIKE „PTRN“*.

Šablonas gali būti sudarytas iš bet kokių simbolių, sudarytų iš abėcėlės, skaičių, specialiųjų simbolių, ir procento ženklo (%) taip pat pabraukimo (_) pakeitimo simbolių, skirtų nesvarbioms žodžio vietoms. Tačiau ši architektūra neleidžia paieškos pagal žodžio ilgį. Todėl šios duomenų bazės projektavimas gali būti pagerintas pridendant papildomą lauką į lentelę word (Pav. 26).



26 Pav. Duomenų bazės projektavimas Nr. 2.

Šis duomenų bazės projektavimo patobulinimas leidžia dvi papildomas užklausas, skirtas atlikti paieškai:

- Parinkti visus atitinkančius žodžius iš tam tikros duomenų bazės lentelės vehicleParts duomenų bazėje pagal nurodytą šabloną PTRN kur žodžio ilgis yra tiksliai N. Pvz: *SELECT word.word FROM word WHERE word.part_id = vehicleParts.part_id AND word.number_of_letters = N*.
- Parinkti visus atitinkančius žodžius iš tam tikros duomenų bazės lentelės vehicleParts duomenų bazėje pagal nurodytą šabloną PTRN kur žodžio ilgis yra daugiau nei N1 ir mažiau nei N2. Pvz: *SELECT word.word FROM word WHERE word.part_id = vehicleParts.part_id AND word.word LIKE 'PTRN' AND (word.number_of_letters > N1 AND word.number_of_letters < N2)*.

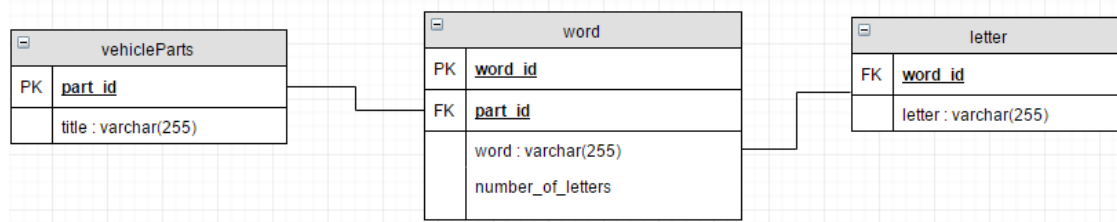
Gali būti skirtingų žodžio ilgio apribojimų kombinacijų, tačiau dažniausiai naudojamos šios dvi. SQL užklausa, skirta gauti visus žodžius su tiksliu ilgiu, taip pat gali būti realizuota ir su pirmuoju duomenų bazės projektavimu. Tam tikslui turėtų būti naudojamas.

pabraukimo pakaitos simbolis (_), kad identifikuoti visas nežinomas raides ir jų poziciją žodyje. Tačiau norint nurodyti žodžio raidžių intervalą pirmasis projektavimas netinka.

Norint suprojektuoti sistemą su didele simbolių eilučių paieškos algoritmų įvairove naudojama krapštymo tipo (angl. scrabble-type) paieška. Šis paieškos tipas dažniausiai neturi šablono, o tik galimų raidžių sąrašą, kuris naudojamas, kad gauti žodį. Tai sunku padaryti su prieš tai aprašytu duomenų bazės projektavimo variantu: turi būti sugeneruotos visos galimos žinomų raidžių

kombinacijos; didelis skaičius galimų žinomų raidžių kombinacijų gali būti sugeneruotos ilgam raidžių rinkiniui; SQL nepajėgus generuoti visas galimas žinomų raidžių kombinacijas gali būti sugeneruotos iš žinomo raidžių rinkinio.

Problemos sprendimui reikalingas duomenų bazės projektavimo patobulinimas – DB turi aprašyti visus žodžius, suskaidant juos į raides ir susieti jas su atitinkamais žodžiais (Pav. 27).

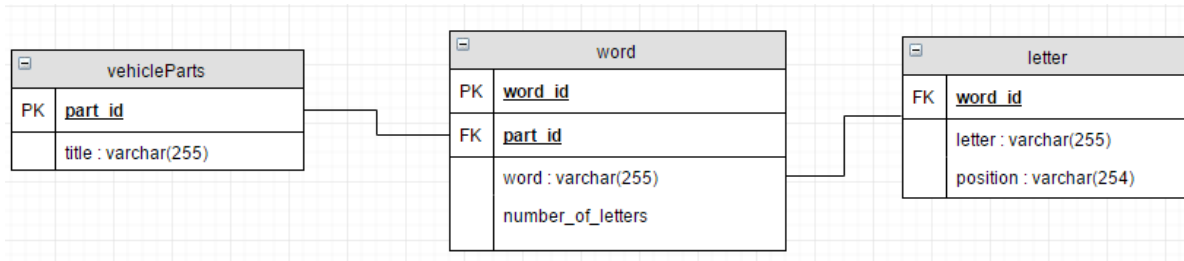


27 pav. Duomenų bazės projektavimas Nr. 3.

Ši architektūra leis surasti žodžius naudojant krapštymo tipo (angl. scrabble-type) žodžių paiešką:

- Atrinkti visus atitinkančius žodžius iš tam tikros duomenų bazės lentelės vehicleParts duomenų bazėje pagal raidžių rinkinį {L1, L2, ..., LN}, kur naudojama mažiausiai M raidžių iš žodžio rinkinio: Pvz. *SELECT word.word FROM letter, word WHERE word.word_id = letter.word_id AND (letter.letter = 'L1' OR letter.letter = 'L2' OR ... OR letter.letter = 'LN') AND word.part_id = vehicleParts.part_id group by letter.word_id HAVING count(letter.word_id) > M.*

Šis projektavimas papildomai gali būti praplėstas žodžių ieškojimui pagal šabloną, naudojant lentelės raidę vietoj žodžio. Reikia tik pridėti raidės poziciją žodyje. (žr. Pav. 28).



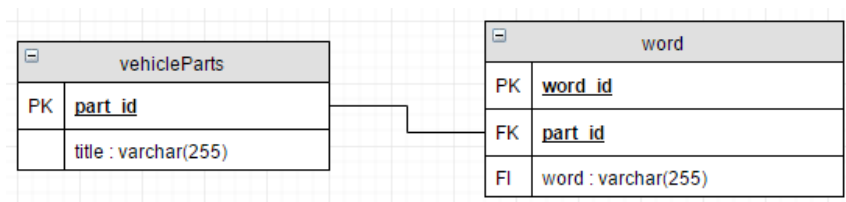
28 pav. Duomenų bazės projektavimas Nr. 4.

- Atrinkti visus atitinkančius žodžius iš tam tikros duomenų bazės lentelės vehicleParts duomenų bazėje pagal raidžių rinkinį {L1, L2, ..., LN} kur žodžio ilgis yra tiksliai M. Pvz. *SELECT word.word FROM letter, word vehicleParts WHERE word.number_of_letters = M AND word.word_id = letter.word_id AND ((letter.letter = „L1“ and letter.position=1) OR (letter.letter = „L2“ AND letter. position=2) OR OR (letter.letter = „LN“ AND letter.*

position=N)) AND word.part_id = vehicleParts.part_id group by letter.word_id HAVING count(letter.word_id) > N.

Paskutinis projektavimo būdas naudojant FULLTEXT indeksavimą – vienas iš FULLTEXT paieškos privalumų – loginės paieškos galimybė – kada yra palaikomi visi pagrindiniai loginiai operatoriai, taip pat toks paieškos būdas siūlo ir daugiau privalumų:

- Nereikia papildomo programavimo kuriant paieškos sistemą, realizuojant indeksavimą.
- Realaus laiko paieška (įrašai indeksuojami juos įterpant).
- Loginių užklausų palaikymas.
- Panašių rezultatų atrinkimas (QUERY EXPANSION).



29 pav. Duomenų bazės projektavimas Nr. 5.

- Atrinkti visus atitinkančius žodžius iš duomenų bazės pagal šabloną PTRN. Pvz: *SELECT word.word FROM word WHERE MATCH (word) AGAINST('PTRN')*
- Atrinkti visus atitinkančius žodžius iš tam tikros duomenų bazės lentelės duomenų bazėje pagal šabloną PTRN. Pvz: *SELECT word.word FROM word WHERE word.dictionary_id = DICT AND MATCH (word) AGAINST('PTRN')*
- Raktinių žodžių nuspėjimas, yra galimybė nuspėti susijusius rezultatus, nebūtinai turinčius ieškomus žodžius PTRN. Pvz: *SELECT word.word FROM word WHERE MATCH (word) AGAINST('PTRN')*

SELECT word.word FROM word WHERE MATCH (word) AGAINST('PTRN', WITH QUERY EXPANSION)

Įvykdžius tokią užklausą bus atrinkti žodžiai turintys „PTRN“, pakartotinai vykdant užklausą su QUERY EXPANSION parametru, bus atrinkti galimai susiję įrašai. Įrašų susiejimas matuojamas ieškant bendrų žodžių. Pavyzdžiui, jei tarp po pirmos užklausos vykdymo gautų rezultatų dažnai buvo rastas žodis „Generatorius“, tai po antros užklausos vykdymo bus pateikti rezultatai turintys šį raktinį žodį, tačiau nebūtinai turintys žodį „PTRN“

- Atrinkti visus atitinkančius žodžius iš tam tikros duomenų bazės lentelės duomenų bazėje pagal žodį „PTRN“, tačiau paieškos rezultatuose būtinai negali būti žodžio „generatorius“ Pvz: *SELECT word.word FROM word vehicleParts WHERE word.part_id =*

vehicleParts.part_id AND MATCH (word) AGAINST('+PTRN -generatorius' IN BOOLEAN MODE)

7.3. ŽODŽIO PAIEŠKOS ALGORITMŲ RELIACINĖMIS DUOMENŲ BAZĖMIS TYRIMAS

Norint nustatyti geriausią DB architektūrą iš ankščiau pasiūlytų, reikia atlikti jų efektyvumo tyrimą. Tam tikslui MySQL duomenų bazių valdymo sistemoje sukuriamos penkios skirtingos DB architektūros. Tikrinant šių DB architektūrų efektyvumą, atliekami eksperimentai su skirtingu žodžių skaičiumi (100, 1000, 5000 žodžių DB). Visos SQL užklausos testuojamos naudojant skirtingus šablonus ar raidžių rinkinius.

Tyrimo tikslas būtų įvertinti užklausos vykdymo laiką kuris reikalingas gauti paieškos rezultatus. Testavimas bus atliekamas su atsitiktiniais žodžiais iš anglų kalbos žodyno. Bus atliekamos dviejų tipų užklausos (sudėtinga ir paprasta) su kiekvieno projektavimo duomenų baze ir vidutinis užklausos laikas bus palyginamas su kito projektavimo duomenų bazėmis.

7.3.1. Duomenų bazės projektavimo Nr. 1 tyrimas

Norint ištestuoti pirmą DB projektavimą, naudotos 4 SQL užklausos, naudojamos gauti visus žodžius, kurie prasideda su raide „a“ skirtingose užklausose:

1. SELECT word.word FROM word WHERE word.word LIKE 'a%'
2. SELECT word.word FROM word WHERE word.word LIKE 'a_%'
3. SELECT word.word FROM word WHERE word.part_id = 1 AND word.word LIKE 'a%'
4. SELECT word.word FROM word WHERE word.part_id = 1 AND word.word LIKE 'a_%'

Testo rezultatai pateikti pirmoje lentelėje.

46 lentelė. Pirmojo duomenų bazės projektavimo bandymo rezultatai.

| Įrašai duomenų bazėje | Užklausos įvykdymo laikas, Ms. | | | | Grajintų eilučių skaičius | | | |
|-----------------------|--------------------------------|------------|------------|------------|---------------------------|------------|------------|------------|
| | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa |
| 100 eilučių | 1 | 1 | 1 | 1 | 100 | 90 | 100 | 90 |
| 1000 eilučių | 2 | 2 | 2 | 2 | 1000 | 990 | 1000 | 990 |
| 10000 eilučių | 6 | 6 | 6 | 6 | 4080 | 4070 | 4080 | 4070 |
| 50000 eilučių | 8 | 8 | 8 | 8 | 4080 | 4070 | 4080 | 4070 |

Kaip rodo testo rezultatai, įvykdymo laikas priklauso nuo žodžių skaičiaus duomenų bazėje. Tačiau, SQL užklausų tipai neturi didelio poveikio greičiui, bet rastų elementų skaičius skiriasi priklausomai nuo naudojamų užklausų.

7.3.2. Duomenų bazės projektavimo Nr. 2 tyrimas

Antrojo duomenų bazės projektavimo testavimas atliktas kaip ir pirmojo, tačiau naudotas žodžio ilgio apribojimas:

1. SELECT word.word FROM word WHERE word.word LIKE 'a%' AND word.number_of_letters = 7
2. SELECT word.word FROM word WHERE word.word LIKE 'a_%' AND (word.number_of_letters > 3 AND word.number_of_letters < 8)
3. SELECT word.word FROM word WHERE word.part_id = 1 AND word.word LIKE 'a%' AND word.number_of_letters = 7
4. SELECT word.word FROM word WHERE word.part_id = 1 AND word.word LIKE 'a_%' AND (word.number_of_letters > 3 AND word.number_of_letters < 8)

Testo rezultatai pateikti antroje lentelėje, kur matomos panašios tendencijos. Tačiau pastebimas užklauso įvykdymo laiko sumažėjimas. Tai reiškia, kad žodžio ilgio saugojimas gali optimizuoti SQL užklauso įvykdymo laiką žodžio paieškoje, kai žinomas norimas žodžio ilgis.

47 lentelė. Antrojo duomenų bazės projektavimo bandymo rezultatai.

| Įrašai duomenų bazėje | Užklauso įvykdymo laikas, Ms. | | | | Gražintų eilučių skaičius | | | |
|-----------------------|-------------------------------|------------|------------|------------|---------------------------|------------|------------|------------|
| | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa |
| 100 eilučių | 1 | 1 | 1 | 1 | 30 | 50 | 30 | 50 |
| 1000 eilučių | 1 | 1 | 1 | 1 | 110 | 380 | 110 | 380 |
| 10000 eilučių | 1 | 3 | 1 | 3 | 510 | 1860 | 510 | 1860 |
| 50000 eilučių | 5 | 6 | 5 | 6 | 510 | 1860 | 510 | 1860 |

7.3.3. Duomenų bazės projektavimo Nr. 3 tyrimas

Trečiasis projektavimas turi daugiau skirtumų palyginus su pirmuoju, kadangi turi papildomą lentelę, kur saugoma abėcėlė, skaičiai ir kt. spec. simboliai kurie yra galimi paieškoje. Todėl naudojamos daugialypės SQL užklauso, kad būtų sujungta kelios sudėtinės lentelės. Tačiau veiktų ir tos pačios užklauso kurios buvo naudotos dvejuose pirmuose bandymuose. Trečio bandymo užklauso:

1. SELECT word.word FROM word WHERE word.part_id = 1 AND word.word LIKE 'a%' AND word.number_of_letters = 7
2. SELECT word.word FROM word WHERE word.part_id = 1 AND word.word LIKE 'a_%' AND (word.number_of_letters > 3 AND word.number_of_letters < 8)
3. SELECT word.word FROM letter, word WHERE word.word_id = letter.word_id AND (letter.letter = 'a' OR letter.letter = 'b') AND word.part_id = 1 group by letter.word_id HAVING count(letter.word_id) > 1

4. `SELECT word.word FROM letter, word WHERE word.word_id = letter.word_id AND (letter.letter = 'a' OR letter.letter = 'b' OR letter.letter = 'r') AND word.part_id = 1 group by letter.word_id HAVING count(letter.word_id) > 3`

Kaip matosi išbandymo rezultatų (žr. 48 lentelę) sudėtingų užklausų įvykdymo laikas yra ilgesnis palyginus su užklausomis, kurios nenaudoja lentelės „word“. Kitas įdomus dalykas yra tas, kad palyginus 3 ir 4 užklausų įvykdymo laikus, kai DB yra 10000 žodžių – 4 užklausa yra greitesnė kai yra iki 10000 žodžių, o 3 užklausa greitesnė, kai žodyne yra 50000 žodžių. Tai veda prie fakto, kad užklausų greitis priklauso nuo žodžio duomenų bazėje.

48 lentelė. Trečiojo duomenų bazės projektavimo bandymo rezultatai.

| Įrašai duomenų bazėje | Užklausos įvykdymo laikas, Ms. | | | | Gražintų eilučių skaičius | | | |
|-----------------------|--------------------------------|------------|------------|------------|---------------------------|------------|------------|------------|
| | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa |
| 100 eilučių | 1 | 1 | 1 | 1 | 30 | 50 | 90 | 20 |
| 1000 eilučių | 1 | 1 | 2 | 1 | 110 | 380 | 550 | 60 |
| 10000 eilučių | 2 | 2 | 8 | 6 | 5100 | 1860 | 3540 | 520 |
| 50000 eilučių | 4 | 5 | 22 | 26 | 5100 | 1860 | 6600 | 1110 |

7.3.4. Duomenų bazės projektavimo Nr. 4 tyrimas

Ketvirtojo projektavimo eksperimentui naudotos keturios skirtingos SQL užklausos:

1. `SELECT word.word FROM word WHERE word.part_id = 1 AND word.word LIKE 'a%' AND word.number_of_letters = 7`
2. `SELECT word.word FROM word WHERE word. part_id = 1 AND word.word LIKE 'a_%' AND (word.number_of_letters > 3 AND word.number_of_letters < 8)`
3. `SELECT word.word FROM letter, word WHERE word.number_of_letters = 7 AND word.word_id = le-tter.word_id AND ((letter.letter = 'a' and letter.position=1) OR (letter.letter = 'b' AND letter.position=2)) AND word.part_id = 1 group by letter.word_id HAVING count(letter.word_id) > 1`
4. `SELECT word.word FROM letter, word WHERE word.number_of_letters = 7 AND word.word_id = letter.word_id AND ((letter.letter = 'a' and letter.position=1) OR (letter.letter = 'b' AND letter.position=2) OR (letter.letter = 'i' AND letter.position=3)) AND word.part_id = 1 group by letter.word_id HA-VING count(letter.word_id) > 2`

Šis duomenų bazės projektavimas leidžia ieškoti žodžių su labai specifiniu raidžių ar simbolių šablonu. Taip pat užklausų įvykdymas (žr. 49 lentelę) yra žymiai ilgesnis. Tai veda prie fakto, kad šis paieškos tipas gali būti naudingas tam tikrose situacijose tačiau jis neturėtų būti naudojamas paprastai žodžių paieškai, kadangi pailgins bendrą sistemos paieškos laiką.

49 lentelė. Ketvirtojo duomenų bazės projektavimo testavimo rezultatai.

| Įrašai duomenų bazėje | Užklausoje įvykdymo laikas, Ms. | | | | Gražintų eilučių skaičius | | | |
|-----------------------|---------------------------------|------------|------------|------------|---------------------------|------------|------------|------------|
| | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa |
| 100 eilučių | 1 | 1 | 1 | 1 | 30 | 50 | 30 | 1 |
| 1000 eilučių | 1 | 1 | 4 | 5 | 110 | 380 | 40 | 1 |
| 10000 eilučių | 1 | 2 | 42 | 65 | 510 | 1860 | 40 | 1 |
| 50000 eilučių | 5 | 6 | 350 | 520 | 510 | 1860 | 40 | 1 |

7.3.5. Duomenų bazės projektavimo Nr. 5 tyrimas

Norint ištestuoti penktąjį DB projektavimą, buvo panaudotos 4 SQL užklausoje, naudojamos gauti visus žodžius, kurie atitinka žodį „generatorius“ skirtingose užklausoje:

1. `SELECT word.word FROM word WHERE MATCH (word) AGAINST('generatorius')`
2. `SELECT word.word FROM word WHERE word.part_id = 1 AND MATCH (word) AGAINST('generatorius')`
3. `SELECT word.word FROM word WHERE MATCH (word) AGAINST('generatorius', WITH QUERY EXPANSION)`
4. `SELECT word.word FROM word WHERE word.part_id = 1 AND MATCH (word) AGAINST('+generatorius -dinama' IN BOOLEAN MODE)`

Bandymo rezultatai pateikti 50 lentelėje.

50 lentelė. Penktojo duomenų bazės projektavimo testavimo rezultatai.

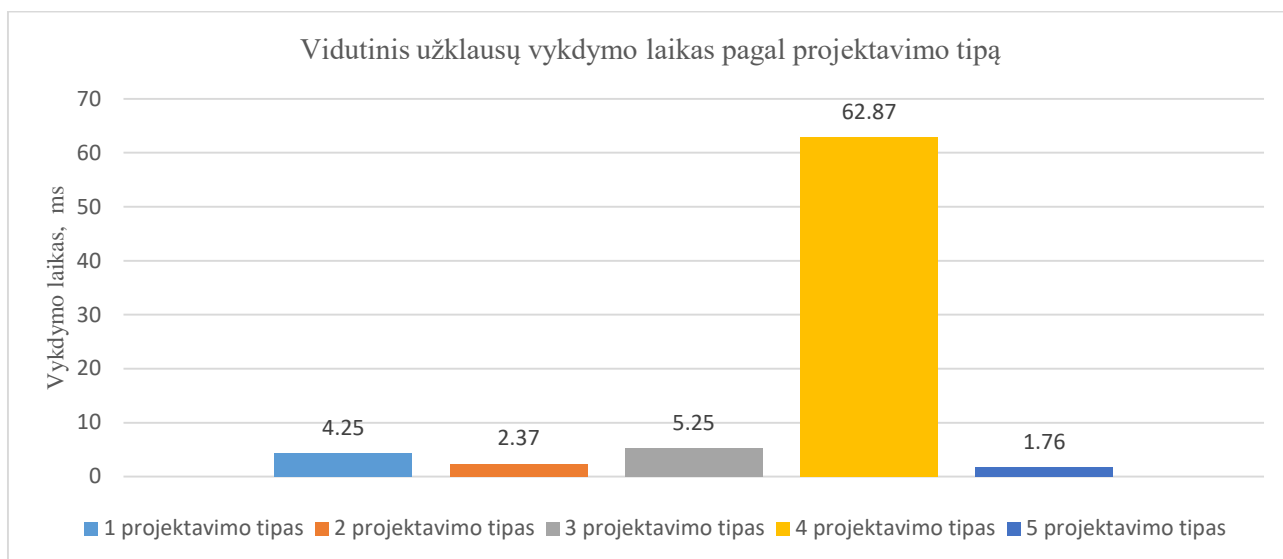
| Įrašai duomenų bazėje | Užklausoje įvykdymo laikas, Ms. | | | | Gražintų eilučių skaičius | | | |
|-----------------------|---------------------------------|------------|------------|------------|---------------------------|------------|------------|------------|
| | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa | 1 užklausa | 2 užklausa | 3 užklausa | 4 užklausa |
| 100 eilučių | 0,3 | 0,3 | 0,5 | 0,7 | 5 | 5 | 25 | 3 |
| 1000 eilučių | 0,3 | 0,9 | 1,5 | 1 | 7 | 10 | 30 | 6 |
| 10000 eilučių | 0,5 | 1,5 | 2 | 3 | 12 | 25 | 69 | 10 |
| 50000 eilučių | 0,8 | 2 | 3 | 4 | 27 | 34 | 102 | 17 |

Išanalizavus penktojo projektavimo tipo rezultatus, kada yra naudojamas indeksavimas matyti, kad visos užklausoje yra vykdomos akiviazdžiai greičiau palyginus su prieš tai atliktais DB projektavimo bandymais. Nors indeksų nauda yra akivaizdi atliekant paiešką DB, bet yra ir keletas neigiamų faktorių, kuriuos svarbu paminėti. Visų pirmą indeksai užima papildomą vietą. Aišku ta papildoma vieta diske yra labai stipriai kompensuojama paieškų vykdymo greičiu, antra turint daug ir didelių indeksų vienoje lentelėje kartais gali reikšti, kad įterpimo ar atnaujinimo (INSERT, UPDATE ar pan.) operacijos bus lėtos.

7.4. TYRIMO SANTRAUKA

Kaip matome iš tyrimo, laikas per kurį atliekamos paprastos užklausoje, nepriklauso nuo DB struktūros, o priklauso tik nuo to, kiek duomenų yra duomenų bazėje. Daugiau laiko atlikti sudėtingą užklausoje užtruko tada kai DB buvo 50.000 įrašų (4 projektavimo tipas, 4 užklausa).

Vidutinis užklausių įvykdymo laikas priklausomai nuo projektavimo tipo yra parodytas paveikslėlyje žemiau:



30 pav. Vidutinis užklausių vykdymo laikas pagal projektavimo tipą.

Penktasis projektavimo tipas veikia geriausiai dėl jame naudojamo indeksavimo, taip pat galima tiksliai aprašyti užklausą, bei užklausių įvykdymo greitis yra didžiausias. Šis tipas yra pasirinktas ir realizuojant žodinę paiešką kuriamoje sistemoje, kadangi joje akcentuojamas greitas automobilių detalių radimas. Tuo tarpu ketvirtas tipas reikalauja per daug laiko visoms užklausoms įvykdyti ir galima sakyti yra nenaudojamas dėl per ilgo laiko vykdant užklausas.

Labiausiai subalansuotas rezultatas yra trečio tipo, dėl tikslių užklausių, kurios nereikalauja daug laiko įvykdymui.

Nepastebėta jokie sąryšio tarp užklaustos įvykdymo laiko ir gautų rezultatų skaičiaus. Užklaustos vykdymo laikas labiau priklausomas nuo reikiamų patikrinti įrašų skaičiaus. Todėl, efektyvus būdas sumažinti žodžio paieškos vykdymo laiką yra žodžio ilgio apribojimas, kuris sumažina tikrinamų įrašų skaičių duomenų bazėje tuo pačiu sumažindamas ir užklaustos vykdymo laiką. Kitas efektyvus būdas pagreitinti paiešką yra indeksų naudojimas – MySQL FULLTEXT indeksavimas leido padidinti užklausių vykdymo greitį iki 34% lyginant su greičiausiai vykdomomis paieškos užklausomis kurios nenaudoja indeksavimo.

7.5. TYRIMO IŠVADOS

1. Simbolių eilutės paieškos algoritmai gali skirtis savo sudėtingumu ir reikalauti tekstinės analizės, tuo tarpu SQL kalba paremta reliacinė duomenų bazė leidžia lengvą žodžio paiešką pagal žinomą raidžių rinkinio šabloną.
2. Norint suprojektuoti žodžio paieškos algoritmo funkcionalumą reliacinėje duomenų bazėje, gali būti panaudojamos SQL užklausų kalboje esančios struktūros. Todėl tinkamas DB projektavimas yra labai svarbus norint užtikrinti sklandžią žodžio paiešką sistemoje.
3. Paieškos trukmė yra tiesiogiai proporcinga užklausoje esančių žodžių kiekiui. Ši priklausomybė yra tiesinė. Realizuotoje sistemoje, kai duomenys saugomi InnoDB lentelėse paieška vykdoma iki 30% lėčiau, nei tada, kai duomenys saugomi MyISAM.
4. Visi penki siūlomi duomenų bazės projektavimai ir išbandytų SQL užklausų rezultatai duoda teisingus rezultatus (randami visi žodžiai, kurie atitinka eilutės rinkinio šabloną). Todėl pagrindinis DB projektavimo pasirinkimas turėtų būti daromas pagal užklausos vykdymo laiką.
5. Atliekant paiešką kada naudojamas indeksavimas matyti, kad tai labai pagerina užklausos vykdymo laiką, tačiau taip pat egzistuoja tam tikri indeksų apribojimai, pavyzdžiui ribotas indekso ilgis, ribotas/reguliuojamas resursų išskyrimas atliekant veiksmus su indeksais.

8. KOKYBĖS VERTINIMO ATASKAITA

8.1. ĮVADAS

Skyriuje pateikiama kokybės vertinimo ataskaita, kurioje nagrinėjama ar sukurta sistema atitinka visus kokybės reikalavimus – lyginama tai kas buvo planuota atlikti ir kas buvo atlikta. Taip pat skyriuje pateikiami likę neišspręsti klausimai, svarbūs pakeitimai, kurie įvyko sistemos kūrimo metu.

8.2. REALIAI ATLIKTO DARBO KOKYBĖS ANALIZĖS TIKSLAI

8.2.1. Aptikti klaidas funkcionavime, logikoje, realizacijoje

Klaidos programinės įrangos funkcionavime, logikoje, realizacijoje buvo aptiktos naudojantis procedūromis nurodytomis testavimo plane (žr. 7 skyrių).

8.2.2. Patikrinti ar sistema atitinka reikalavimų specifikaciją

Ar programų sistema atitinka reikalavimų specifikaciją buvo patikrinta naudojant priėmimo testavimą, kurio procedūros ir rezultatai pateikti 7 – tame skyriuje.

8.3. KOKYBĖS VERTINIMO PROCESAS.

8.3.1. Peržiūros

Atlikus programavimo darbus ir užbaigus projektą vykdytojų komanda atlieka apžvalgą, kurios tikslas yra surinkti informaciją apie projektą, kas jame pavyko gerai ir kas nepasisėkė, kad tai būtų galima panaudoti sekančiuose projektuose. Ši apžvalga paprastai yra atliekama projekto pabaigoje. Jei projektas didesnis apžvalga gali būti atliekama ir po kiekvienos projekto svarbesnės dalies pabaigimo.

8.3.2. Interviu su užsakovu

Interviu su užsakovu buvo atliekamas visą sistemos kūrimo laikotarpį. Nuolatos aptariami atlikti pakeitimai ir patobulinimai.

8.3.3. Projektavimo komandos narių peržiūrų aprašymas

Peržiūrų metu, buvo aptartos sistemos dalys, kurių kokybė kelia abejonių ar funkcionalumas nėra iki galo įgyvendintas.

8.3.4. Vaidmenys ir atsakomybės

Projekto dalyvių vaidmenys ir atsakomybės:

- Sistemos projektuotojas – Ovidijus Kavaliauskas;
- Sistemos programuotojas – Ovidijus Kavaliauskas;
- Sistemos testuotojas – Ovidijus Kavaliauskas;
- Sistemos analitikas – Vacius Jusas.

8.3.5. Programinio kodo peržiūra

Parašyto kodo peržiūros metu didžiausias dėmesys buvo sutelktas į:

- Ar nėra perdėto parametrų naudojimo generic tipuose;
- Ar naudojami generic tipai, kur galima;
- Ar nėra klaidų kintamųjų, metodų, klasių pavadinimuose;
- Ar klasėse panaudojamos visos įtrauktos klasės;
- Ar tinkamai naudojamas klasių paveldimumas.

8.3.6. Vertinimo rezultatai

Žemiau esančiose lentelėse pateikiami kokybės parametrai, kuriuos atitinka sukurta sistema.

51 lentelė. Peržiūros vertinimo rezultatai.

| Parametras | Ar atitinka |
|--|-------------|
| Ar sistemos kodas atitinka „object-oriented“ principus | Iš dalies |
| Ar kintamųjų, metodų, klasių pavadinimai atitinka konvenciją | Taip |
| Ar tuo pačiu vardu neegzistuoja globalių objektų | Taip |
| Ar deklaruotam objektui priskirta reikšmė | Taip |

52 lentelė. Kokybės parametrai.

| Eil. Nr. | Parametras | Taip/Ne | Aprašymas |
|----------|----------------|---------|--|
| 1. | Iš plečiamumas | Taip | Projektas yra pakankamai dokumentuotas, kode yra komentarai. Kodas parašytas laikantis objektinio programavimo standartų, naudojantis architektūros specifikacija. |
| 2. | Panaudojamumas | Taip | Sistemai sukurta vartotojo dokumentacija, leidžianti naujiems naudotojams lengviau ją suprasti. |
| 3. | Patvarumas | Taip | Visi į sistemą įvedami duomenys yra patikrinami. |

8.3.7. Darbų našumas

Dauguma suplanuotų darbų buvo atlikti laiku pagal grafiką. Likę darbai buvo suderinti su užsakovu ir atlikimo grafikas pakoreguotas.

8.4. IŠVADOS

1. Atliktas kokybės vertinimas parodė, kad sukurta sistema atitinka užsibrėžtus ir tokiai sistemai keliamus kokybės reikalavimus – sistemoje laikomasi „object-oriented“ programavimo principo, kintamųjų, klasių, metodų pavadinimai atitinka konvenciją, neegzistuoja globalių objektų tokiu pačiu pavadinimu. Sistemos kode naudojami komentarai, o pati sistemos dokumentacija yra pakankamai išsami ir leidžianti greitai perprasti visą sistemą.

9. IŠVADOS

1. Esamų sprendimų analizė parodė jog nėra sandėlio valdymo sistemos, kuri būtų specializuota būtent naudotomis automobilių dalimis prekiaujančioms įmonėms, todėl buvo nuspręsta kurti sistemą, kuri būtų šiuolaikiškas, patogus ir nebrangus įrankis skirtas būtent autolaužynų verslui.
2. Atlikus programavimo priemonių analizę buvo pasirinktas Laravel 5 programavimo karkasas ir MySQL 5 duomenų bazių valdymo sistema. Darbe aprašyti Laravel 5 karkaso privalumai prieš kitas technologijas ir geras ryšys su MySQL DBVS nulėmė šį pasirinkimą.
3. Viso projekto vykdymo metu buvo sėkmingai sukurta testavimo metodologija ir testavimo planas. Kurio buvo laikomasi atliekant testavimą. Nors dauguma defektų buvo pašalinta dar programavimo fazės metu. Kitos klaidos kurias aptiko testavimo procedūros, sėkmingai buvo pašalintos vėliau.
4. Sukurta sistema atitinka užsibrėžtus kokybės reikalavimus. Sukurtoje sistemoje buvo realizuota didžioji dalis specifikacijoje apibrėžtų funkcijų.

10. LITERATŪRA

1. Sandėlis paklūsta profesionalams [žiūrėta 2015 11 22], prieiga internete: www.meridian.lt/sandelis-paklusta-profesionalams/
2. Sandėlio valdymo sistema vision [žiūrėta 2015 11 22], prieiga internete: <http://www.equinox.lt/sandelio-valdymas/sandelio-valdymo-sistema-vision/>
3. Tyrimas: įmonės sandėliavimo efektyvumą didina vangiai [žiūrėta 2015 11 22], prieiga internete: http://www.softconsulting.lt/naujienos/tyrimas_imonės_sandeliavimo_efektyvuma_didina_vangiai/
4. QR kodai [Žiūrėta: 2015 11 16]. Prieiga per internetą: <http://www.moksliniai.darbai.lt/referatas/12843/qr-kodai.html>
5. QR kodas, kas tai? [Žiūrėta: 2015 11 16]. Prieiga per internetą: <http://www.web4it.lt/qr-kodas/>
6. 2D kodų naudojimas [Žiūrėta: 2015 11 16]. Prieiga per internetą: www.kvt.lt/lt/qr
7. Žmogaus-kompiuterio sąveika. Taisyklės, principai, rekomendacijos [Žiūrėta: 2015 11 16]. Prieiga per internetą: http://www.mif.vu.lt/~pijus/ZKS/Princip_rekomend.pdf
8. Vartotojo sąsajos projektavimas [Žiūrėta: 2015 11 16]. Prieiga per internetą: http://www.mif.vu.lt/~ragaisis/PSI_mag2008/StudMedziaga/Interfeiso_projektavimas-8gr_MariusRudys.pdf
9. Žiniatinklio taikomųjų programų projektavimas [Žiūrėta: 2015 11 16]. Prieiga per internetą: http://proin.ktu.lt/~webdev/index.php?option=com_k2&view=item&id=15:taikomuju-programu-projektavimas&tmpl=component&print=1&Itemid=80
10. Model–view–controller: A Design Pattern for Software [Žiūrėta: 2015 11 16]. Prieiga per internetą: <https://ist.berkeley.edu/as-ag/pub/pdf/mvc-seminar.pdf>
11. Model-View-Controller (MVC) Architecture [Žiūrėta: 2015 11 16]. Prieiga per internetą: <http://www.battersea-locksmith.co.uk/briefings/MVC.pdf>
12. MVC procesas ir architektūrą Architecture [Žiūrėta: 2015/11/16]. Prieiga per internetą: <http://pixel.lt/tag/mvc>
13. Web sistemų projektavimas ir testavimas [Žiūrėta: 2015 11 17]. Prieiga per internetą: http://webcache.googleusercontent.com/search?q=cache:bxNay9vf2WsJ:www.mif.vu.lt/~ragaisis/PSI_mag2008/StudMedziaga/Web_sistemu_projektavimas%26testavimas-7gr_NikolajGrigorjev.doc+&cd=1&hl=lt&ct=clnk&gl=us
14. Web aplikacijų testavimo veiklos tobulinimas [Žiūrėta: 2015 11 17]. Prieiga per internetą: http://leidykla.vgtu.lt/conferences/jmk_informatika_2008/files/pdf/kuliesius_32-38.pdf
15. Testing a Website: Best Practices [Žiūrėta: 2016 11 19]. Prieiga per internetą: http://www.cmcrossroads.com/sites/default/files/article/file/2013/XUS893545file1_0.pdf

16. R. Rivest, 1977, On the worst-case behavior of string-searching algorithms. [Žiūrėta: 2017 02 02]. Prieiga per internetą: <http://orion.lcg.ufrj.br/Dr.Dobbs/books/book5/chap10.htm>
17. R. Baeza-Yates, 1989, Improved string searching. [Žiūrėta: 2017 02 02]. Prieiga per internetą: <http://onlinelibrary.wiley.com/doi/10.1002/spe.4380190305/full>
18. D.E. Knuth, J. Morris, and V. Pratt. Fast pattern matching in strings. [Žiūrėta: 2017 02 02]. Prieiga per internetą: <https://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/Knuth77.pdf>
19. R. Boyer and S. Moore, 1977, A fast string searching algorithm. [Žiūrėta: 2017 04 02]. Prieiga per internetą: <http://www.jpk.pku.edu.cn/pkujpk/course/sjg/chapter3/resource/A%20Fast%20String%20Searching%20Algorithm.pdf>
20. Jeffrey D. Ullman, [Žiūrėta: 2017 04 19]. Prieiga per internetą: <http://infolab.stanford.edu/~ullman/focs/ch08.pdf>
21. Okunis R., Makarevičius M. Populiariausios duomenų bazių valdymo sistemos.
22. Paliulis E., Baronienė R., 2010, Duomenų bazių valdymo sistemų analizė. Jaunųjų mokslininkų darbai. [Žiūrėta: 2017 04 19]. Prieiga per internetą: <http://gs.elaba.lt/object/elaba:6106131/6106131.pdf>
23. An introduction to the SQL procedure. [Žiūrėta: 2017 04 19]. Prieiga per internetą: <http://www.ats.ucla.edu/stat/sas/library/nesug99/bt082.pdf>

11. PRIEDAI

11.1. 1 PRIEDAS – VARTOTOJO DOKUMENTACIJA

11.1.1. Pagrindinės sistemos funkcijos

Sistema turi šias pagrindines funkcijas:

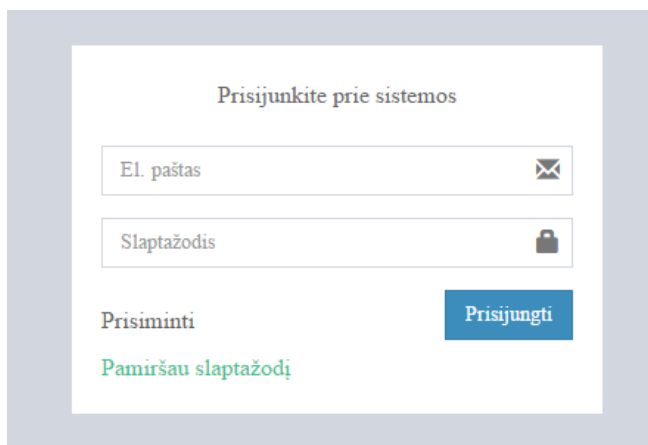
- Darbuotojų valdymas
- Ardomų automobilių valdymas
- Paruoštų detalių pardavimui valdymas
- Detalių nuardymo užsakymų valdymas
- Detalių ir automobilių paieška.
- Detalių kategorijų medžio valdymas

11.1.2. Naudotojo atmintinė

Šiame skyriuje pateikiami pagrindiniai naudotojo veiksmai kurie reikalingi norint naudotis pagrindinėmis sistemos funkcijomis.

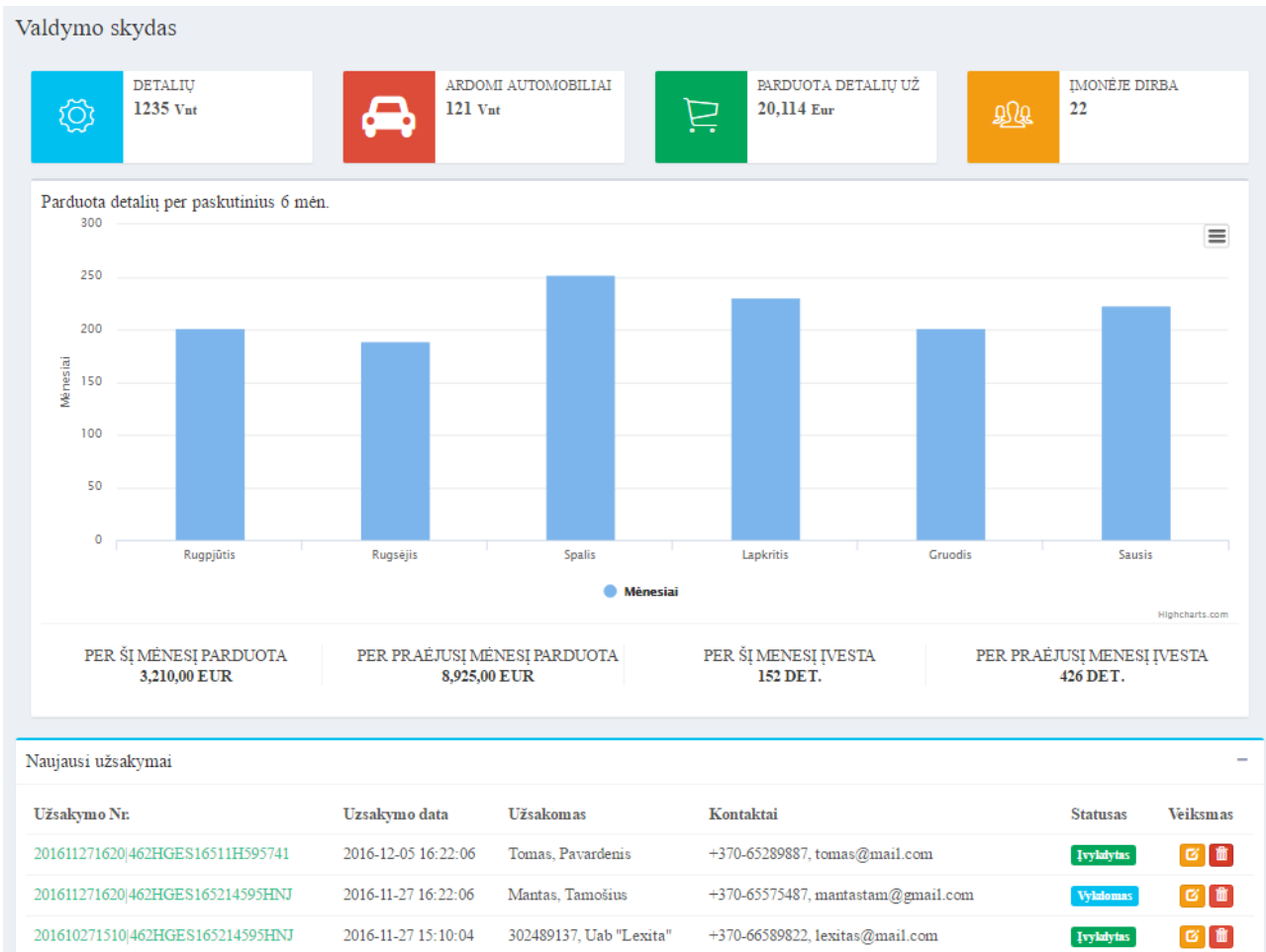
11.1.3. Prisijungimas prie sistemos

Kad pradėti naudotis sistema administratorius jums turi pateikti prisijungimo prie sistemos duomenis. Pirmą kartą jungiantis prie sistemos bus pateiktas prisijungimo vardas ir laikinas slaptažodis (kurį vėliau rekomenduojama pasikeisti). Įsijungus sistemą bus pateikiamas langas (30 pav.) kuriame įvedami jums pateikti duomenys ir paspaudžiamas mygtukas Prisijungti. Po šių veiksmų naudotojui pateikiamas pagrindinis sistemos langas (31 pav.)



31 pav. Prisijungimas prie sistemos.

Pagrindiniame sistemos lange pateikiama įvairi statistinė informacija bei vėliausiai įvestų detalių ir užsakymų sąrašai.



32 pav. Pagrindinis sistemos langas.

11.1.4. Naujo automobilio įvedimas į sistemą

Norint į sistemą pridėti naują automobilį iš meniu reikia pasirinkti *Įvedimas* -> *Automobilis*. Po šių veiksmų atsivers automobilio pridėjimo forma (32 pav.). Kairėje pusėje esančiame kategorijų medyje reikia pasirinkti ir pažymėti visas automobilyje esančias dalis, atlikus šį veiksmą paspaudžiamas mygtukas *Patvirtinti pasirinkimą*. Toliau dešinėje pusėje esančioje formoje pasirenkami įvedamo automobilio duomenys, žemiau esantis langas *Automobilio nuotraukos* leidžia pridėti automobilio nuotraukas – paspaudus mygtuką *pasirinkite nuotraukas* pasirodo standartinis *windows* failų pasirinkimo langas, jame reikia nurodyti kelią iki nuotraukų esančių kompiuteryje ir paspausti mygtuką *select*. Po šių veiksmų atsiras pasirinktų nuotraukų miniatiūros (33 pav.). Jeigu pasirinkote norimas nuotraukas paspauskite mygtuką *pridėti nuotraukas* ir šios bus nusiųstos į serverį. Kitu atveju galima pašalinti nuotrauką ar nuotraukas nuspaudus mygtuką *Pašalinti* esantį po nuotraukos miniatiūra ir pakartotinai atlikti nuotraukų pridėjimo veiksmus. Atlikus visus aukščiau aprašytus veiksmus paspaudžiamas mygtukas *Išsaugoti*, jeigu naudotojas visus veiksmus atliko teisingai jis bus perkeltas į sistemoje esančių automobilių sąrašą ir parodytas pranešimas, kad automobilis sėkmingai įvestas į sistemą. Kitu atveju naudotojo bus paprašyta pataisyti formoje esančias klaidas (34 pav.).

Pridėti automobilį

Pažymėkite automobilyje esančias dalis

- Visas Automobilis
 - Elektronikos Detalės
 - Važiuklės detalės
 - Kėbulo Detalės
 - Variklio detalės
 - Greičiu dezes dalys
 - Radiatoriai
 - Papildomos detalės

Patvirtinti pasirinkimą

Įveskite automobilio parametrus

| | | |
|--|---|--|
| Markė | Modelis | Modifikacija |
| <input type="text" value="Acura"/> | <input type="text"/> | <input type="text"/> |
| Kuro tipas | Kėbulo tipas | Pavarų dėžė |
| <input type="text" value="Pasirinkite kuro tipą"/> | <input type="text" value="Pasirinkite kėbulo tipą"/> | <input type="text" value="Pasirinkite pavarų dėžę"/> |
| Transmisijos tipas | Vairo padėtis | Rida (km) |
| <input type="text" value="Pasirinkite transmisijos tipą"/> | <input type="text" value="Pasirinkite vairo padėtį"/> | <input type="text"/> |
| Kėbulo numeris | | Įsigijimo kaina |
| <input type="text"/> | | <input type="text"/> |

Komentaras

Automobilis neturi aprašymo

Automobilio nuotraukos: [Pridėti nuotraukas](#)

[Išsaugoti](#)

33 pav. Automobilio įvedimo forma.

Automobilio nuotraukos:

[Pridėti nuotraukas](#)



[Išsaugoti](#)

34 pav. Pasirinktos automobilio nuotraukos.

- Pasirinkite kuro tipą
- Pasirinkite kėbulo tipą
- Pasirinkite pavarų dėžės tipą
- Pasirinkite transmisijos tipą
- Pasirinkite vairo padėtį
- Įveskite automobilio kėbulo numerį

35 pav. Pranešimas apie netinkamai užpildytą automobilio įvedimo formą.

11.1.5. Naujos detalės įvedimas į sistemą

Naujos detalės įvedimui iš pagrindinio meniu reikia pasirinkti *Įvedimas -> Detalė*. Po šio veiksmo atsivers detalės įvedimo forma (35 pav.). Pirmiausiai formoje reikia pasirinkti detalės tipą (Ardomo automobilio detalė) arba (Nežinomo automobilio detalė), priklausomai nuo pasirinkimo bus pateikta viena iš formų tolimesniam pildymui. Nepriklausomai nuo detalės tipo kairėje formos pusėje iš kategorijų medžio reikia pasirinkti pridedamą detalę (vienu metu galima pasirinkti tik vieną konkrečią detalę pvz. *Starteris*) ir paspausti mygtuką *Patvirtinti pasirinkimą*. Toliau pildoma dešinėje pusėje pateikta forma. Užpildžius visus laukus galima pridėti detalės nuotraukas, nuotraukų pridėjimas vykdomas taip pat kaip ir automobilio nuotraukų pridėjimas kuris jau buvo aprašytas 2.2 skyriuje. Baigus pildyti formą paspaudžiamas vienas iš dviejų mygtukų *Išsaugoti* – detalė bus įtraukta į sistemą ir apie tai pranešimu informuotas naudotojas arba *Išsaugoti ir spausdinti etiketę* – detalė bus įtraukta į sistemą ir sugeneruota detalės etiketė paruošta spausdinimui (30 pav.). Atspausdinta etiketė prikabinama prie detalės ir padedama sandėlyje.

Pridėti detalę

Pažymėkite pridedamą detalę

- Visas Automobilis
 - Elektronikos Detalės
 - Rankinio stabdžio kompiuteris
 - Zibinto aukscio reguliatorius (korektorius)
 - Navigacijos cd skaitytuvas
 - Liuko varikliukas
 - Variklio kompiuteris
 - Automagnetola
 - ESP greitejimo sensorius
 - Velenelio fazes vostuvas
 - Baterija - Akumulatorius
 - Zemu dazniu garsiakalbis
 - Lango pakalejo valdymo blokelis
 - Smuginis kuro blokavimo daviklis
 - Duru uzrakto mygtukas
 - Vairo stiprintuvo kompiuteris
 - Uzvedimo komplektas - variklio kompiuteris
 - Kuro bako siurblys
 - Centrine vakuumo valdymo pompa
 - Imobilaizerio antena
 - Langu apiplovimo varikliukas
 - Komforto blokas
 - Kuro bako voztuvas
 - Greiciu dezės kompiuteris
 - Lauko temperaturos daviklis
 - Rezistorius

Įveskite detalės duomenis

Ardomo automobilio detalė [Nežinomo automobilio detalė](#)


Automobilis
2HGES16511H595741 Audi A6 1997 - 2004 1.8T (180Hp)

Detalės kodas: Nenustatyta
Detalės kaina: 26

Vieta sandėlyje: VL1-001
Pagaminimo metai: 2005

Komentaras
Kompiuteris yra Siemens gamintojo

Detalės nuotraukos: [Pridėti nuotraukas](#)



[Išsaugoti](#) [Išsaugoti ir spausdinti etiketę](#)

36 pav. Užpildyta detalės įvedimo forma.



37 pav. Etiketės spausdinimo langas.

11.1.6. Paieška sistemoje

Paieškos atlikimui pagrindiniame meniu paspaudžiame mygtuką *Paieška*, kad atsivėrusioje paieškos formoje (36 pav.) atlikti paiešką iš kairėje esančio detalių medžio privaloma pasirinkti ieškomą detalę paspausti mygtuką *Patvirtinti pasirinkimą* ir dešinėje pusėje pasirinkti privalomus paieškos parametrus (Markė, Modelis, Modifikacija). Norint patikslinti paiešką galima pasirinkti ir vieną ar visus paieškos parametrus kurie pateikti skiltyje *Papildomi paieškos parametrai*. Atlikus šiuos veiksmus spaudžiamas mygtukas *Ieškoti*. Po kelių sekundžių sistema pateiks paieškos rezultatus (37 pav.). Paieškos rezultatai yra suskirstyti į *Detalės sandėlyje* (detalės yra sandėlyje ir jas galima parduoti) ir *Automobiliai* (ieškoma detalė yra automobilyje tačiau ją dar reikia nuardyti). Paieškos kriterijų pakeitimui, paieškos rezultatų lange paspauskite *Keisti paiešką* jūs būsite nukreipti į paieškos formą su jūsų anksčiau pasirinktais paieškos parametrais. Norint atlikti paiešką iš naujo paspauskite mygtuką *Nauja paieška*.

Detalių paieška

Pasirinkite ieškomą detalę(-es)

- ▣ Visas Automobiliai
 - ▣ Elektronikos Detalės
 - ▣ Važiuklės detalės
 - ▣ Kėbulo Detalės
 - ▣ Galinės kėbulo detalės
 - ▣ Sparnu moldingai
 - ▣ Sparnu praplatinimai (arkos apdaila)
 - ▣ Salonas
 - ▣ Stogas
 - ▣ Priekinės kėbulo detalės
 - ▣ Slenksčiai
 - ▣ Remas
 - ▣ Durų komplektas
 - ▣ Variklio detalės
 - ▣ Greičiu dezes dalys
 - ▣ Radiatoriai
 - ▣ Papildomos detalės

Patvirtinti pasirinkimą

Privalomi paieškos kriterijai:





| | | |
|-----------------------------------|---|---|
| Markė | Modelis | Modifikacija |
| <input type="text" value="Audi"/> | <input type="text" value="A6 2004 - 2011"/> | <input type="text" value="2.7 TDI (180 Hp)"/> |

Papildomi paieškos kriterijai:

| | | |
|--|---|--|
| Kuro tipas | Kėbulo tipas | Pavarų dėžė |
| <input type="text" value="Pasirinkite kuro tipą"/> | <input type="text" value="Pasirinkite kėbulo tipą"/> | <input type="text" value="Pasirinkite pavarų dėžę"/> |
| Transmisijos tipas | Vairo padėtis | Detalės kodas |
| <input type="text" value="Pasirinkite transmisijos tipą"/> | <input type="text" value="Pasirinkite vairo padėtį"/> | <input type="text"/> |

38 pav. Paieškos forma.

Paieškos rezultatai



| 2 | | 1 | | |
|---|--|-----------------------|-------------|---|
| Detalės sandėlyje | | Automobiliai | | |
| Nuotrauka | Detalė | Gamintojo kodas | Kaina (Eur) | Veiksmai |
|  | Starteris Audi A6 2004 - 2011.7 TDI (180 Hp) | 4d0907379d 0265108005 | 55 |  |
|  | Šviesų jungiklis Audi A6 2004 - 2011.7 TDI (180 Hp) | 4d0907379d 0255770269 | 25 |  |

Keisti paiešką Nauja paieška

39 pav. Paieškos rezultatai.

11.1.7. Užsakymo detalių nuardymui formavimas

Norint suformuoti užsakymą detalių nuardymui nuo automobilio reikia pasirinkti detalę norimo automobilio peržiūrą (38 pav.)

| | | | | | | | |
|---|------|----------------|------------------|-----------|------------|------------|---|
|  | Audi | A6 2004 - 2011 | 2.7 TDI (180 Hp) | Dyzelinas | Priekiniai | Automatinė |  |
|---|------|----------------|------------------|-----------|------------|------------|---|

40 pav. Detali automobilio peržiūra pateikiama paspaudus juodame stačiakampyje esantį mygtuką.

Atsidarius detalios peržiūros langui (39 pav.) kairėje pusėje esančiame detalių medyje reikia pasirinkti norimas nuardyti detales (Detalės kurios dar yra automobilyje pažymėtos žaliu fonu (40 pav.) ir paspausti mygtuką *Užsakyti detales nuardymui*.

Atsidariusiame naujame lange (41 pav.), *Užsakymo informacija* kortelėje pateikiama pagrindinė užsakymo informacija, čia taip pat įrašomos užsakymo pastabos. Kortelė *Kliento informacija* (42 pav.) čia reikia užpildyti kliento kontaktinius duomenis arba tiesiog pasirinkti iš jau esamų jeigu klientas detalių užsakymą vykdo nebe pirmą kartą. Užpildžius visus laukus paspaudžiamas mygtukas *Pateikti užsakymą detalių nuardymui*, po šio veiksmo suformuojamas detalių užsakymo aktas ir atidaromas jo spausdinimo langas (43 pav.). Atspausdintas užsakymas pateikiamas darbuotojui atsakingam už automobilių ardymą.

Automobilio informacija


Automobilio detalės

- Visas Automobilis
 - Elektronikos Detalės
 - Rankinio stabdžio kompiuteris
 - Žibinto aukščio reguliatorius (korektorius)
 - Navigacijos cd skaitytuvas
 - Liuko variklius
 - Variklio kompiuteris
 - Automagnetola
 - ESP greitejimo sensorius
 - Velenelio fazes vstuvus
 - Baterija - Akumuliatorius
 - Zemu dažniu garsiakalbis
 - Lango pakalejo valdymo blokelis
 - Smuiginis kuro blokavimo daviklis
 - Duru uzrakto mygtukas
 - Vairo stiprintuvo kompiuteris
 - Uzvedimo komplektas - variklio kompiuteris
 - Kuro bako siurblys
 - Centrine vakuumo valdymo pompa
 - Imobilizierio antena
 - Langu apiplovimo varikliukas
 - Komforto blokas
 - Kuro bako vstuvus
 - Greiciu dezės kompiuteris
 - Lauko temperatūros daviklis
 - Rezistorius
 - CD keitiklis
 - Kiti kompiuteriai
 - Vairo padeties daviklis
 - Elektrinis slėgio vstuvus
 - Kuro bako siurblio matuokle

Automobilio duomenys

| | |
|-----------------------------|--|
| Įvedė | admin Administratorius |
| Markė: | Audi |
| Modelis: | A6 1997 - 2004 |
| Modifikacija: | 3.0 i V6 (220 Hp) |
| Kuro tipas: | Benzinas |
| Kebulo tipas: | Kabrioletas |
| Greičių dėžė: | Automatinė |
| Transmisija: | Visi variantys (4X4) |
| Vairo padėtis: | Kairė |
| Kebulo numeris: | 2HGES16511H595741 |
| Rida: | 54444 (Km) |
| Įsigijimo kaina: | 155 (Eur) |
| Komentaras apie automobilį: | komentaras |

Automobilio nuotraukos



41 pav. Detali automobilio peržiūra.

Automobilio detalės

- Visas Automobilis
 - Elektronikos Detalės
 - Rankinio stabdžio kompiuteris
 - Žibinto aukščio reguliatorius (korektorius)
 - Navigacijos cd skaitytuvas
 - Liuko variklius
 - Variklio kompiuteris
 - Automagnetola
 - ESP greitejimo sensorius
 - Velenelio fazes vstuvus
 - Baterija - Akumuliatorius
 - Zemu dažniu garsiakalbis

42 pav. Automobilyje esančios detalės pažymėtos žaliu fonu, jas galima užsakyti nuardymui.

| | |
|--|---------------------|
| Užsakymo informacija | Kliento informacija |
| Užsakomo Nr. | |
| <input type="text" value="20161216111249/2HGES16511H595741"/> | |
| Užsakymo statusas | |
| <input type="text" value="Vykdomas"/> | |
| Užsakomų detalių sąrašas | |
| <ul style="list-style-type: none">• Rankinio stabdžio kompiuteris• Liuko varikliukas• ESP greitejimo sensorius | |
| Pastabos | |
| <input type="text" value="Nera"/> | |

Atšaukti

Pateikti užsakymą detalių nuardymui

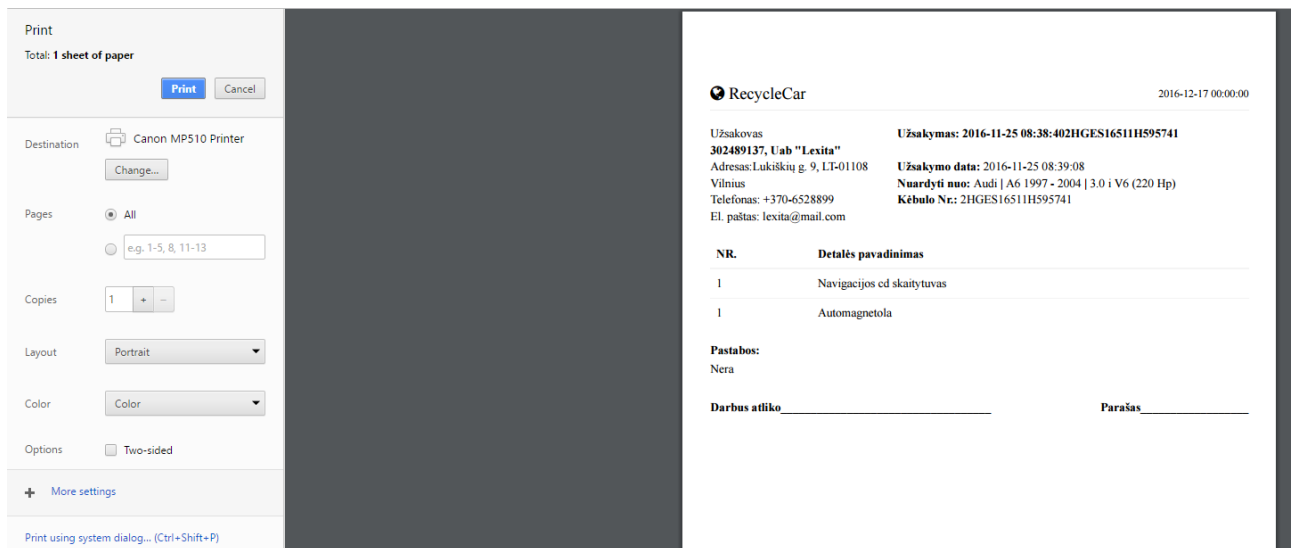
43 pav. Detalių užsakymo nuardymui langas – kortelė (Užsakymo informacija).

| | |
|--|--|
| Užsakymo informacija | Kliento informacija |
| Pastovus klientas (pasirinkite įmonės kodą) | |
| <input type="text" value="Pasirinkite įmonės kodą"/> | |
| Užsakovo vardas (arba įmonės kodas) | El .paštas |
| <input type="text"/> | <input type="text"/> |
| Tel. Nr. | Adresas(pildyti tik juridiniams asmenims) |
| <input type="text"/> | <input type="text"/> |
| Juridinis asmuo: <input type="checkbox"/> | |

Atšaukti

Pateikti užsakymą detalių nuardymui

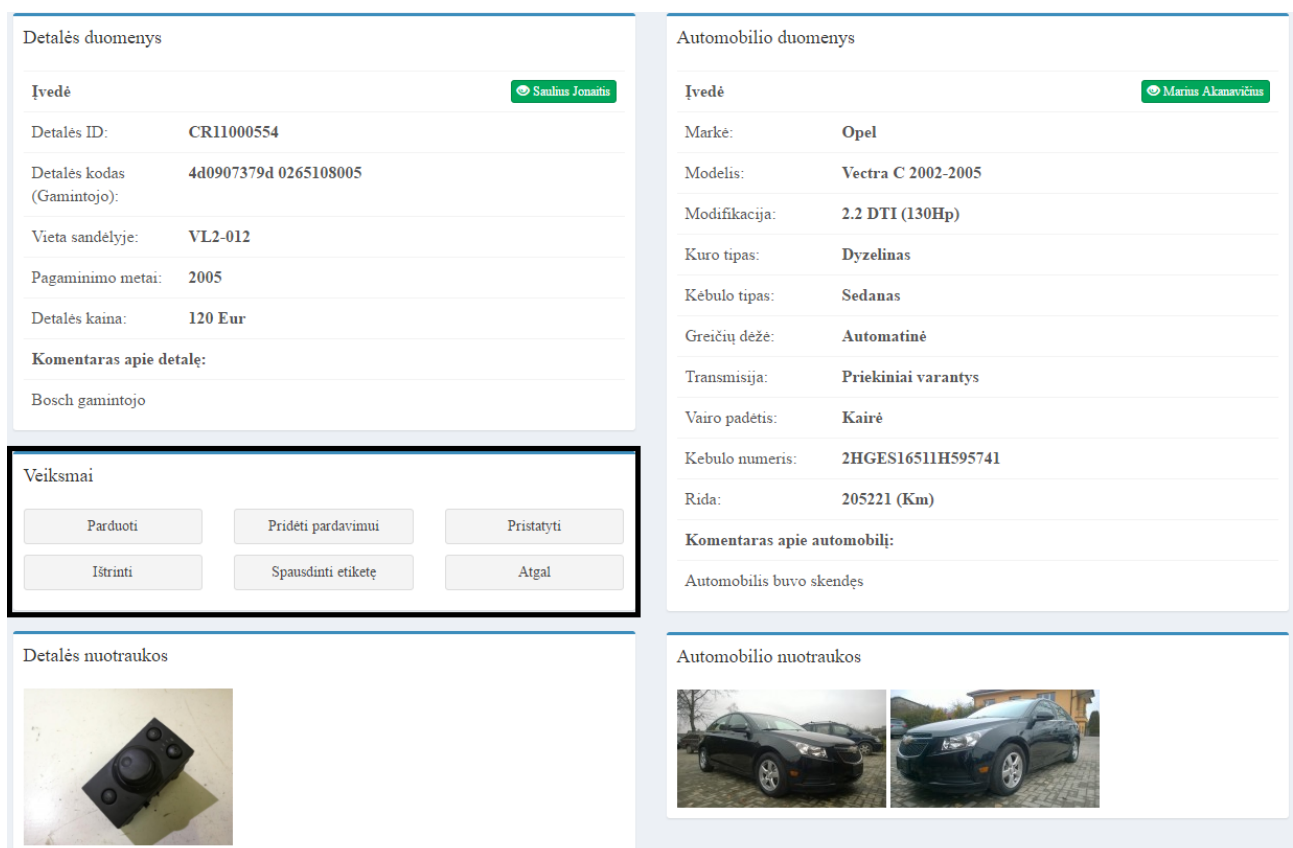
44 pav. Detalių užsakymo nuardymui langas – kortelė (Kliento informacija).



45 pav. Užsakymo spausdinimo langas (detalių nuardymas).

11.1.8. Veiksmai su detale

Norint atlikti veiksmus su pasirinkta detale reikia atsidaryti detalių peržiūros langą (44 pav.), tam naudojamas detalios peržiūros mygtukas .



46 pav. Detali informacija apie detalę.

Parduoti – paspaudus mygtuką atsidaro papildomas detalės pardavimo langas (45 pav.) kuriame pateikiama pagrindinė parduodamos detalės informacija (identifikacijos numeris ir kaina), esant poreikiui galima pasirinkti klientui taikomą nuolaidą. Norint detalę parduoti spaudžiamas

vienas iš dviejų mygtukų *Parduoti* arba *Parduoti ir spausdinti sąskaitą*, pirmuoju atveju detalė bus parduota (pašalinta iš parduodamų detalių katalogo) antruoju papildomai bus suformuota sąskaita faktūra ir pateiktas jos spausdinimo langas (46 pav.).

Pardavimas

Detalės ID: **CR11000554**

Kaina (Eur): **120**

Nuolaida:

Su nuolaida(Eur): **Netaikoma**

Parduoti **Parduoti ir spausdinti sąskaitą**

47 pav. Detalės pardavimo langas.

Print
Total: 1 sheet of paper

Destination: Canon MP510 Printer

Pages: All

Copies: 1

Layout: Portrait

Color: Color

Options: Two-sided

Print using system dialog... (Ctrl+Shift+P)

RecycleCar.EU. 2016-12-15

Pardavėjas: **RecycleCar.EU**
Ryto g. 18, LT-25108
Lentvaris, Trakų raj., Lietuva
Telefonas: +370-656-99987
El. paštas: info@recyclecar.eu

Pristatyti: **Tadas Simonavičius**
Partizanų g. 9, Kaunas
LT-59242, Lietuva
Phone: +370-665-9988874
Email: tadas1155@gmail.com

Užsakymas #007612
Apmokėti iki: 2016-12-29

| NR. | Detalės ID | Detalės pavadinimas | Kaina (Eur) |
|-----|-------------|---------------------|-------------|
| 1 | CR110005544 | ABS kompiuteris | 35 |
| 2 | CR221555522 | Šviesų jungiklis | 15 |

Už prekes: 50 Eur
Pristatymas: 5 Eur
Viso: 55.00 Eur

48 pav. Sąskaitos faktūros spausdinimo langas.

Pridėti pardavimui – mygtukas naudojamas kada klientas pageidauja pirkti daugiau nei vieną detalę, paspaudus mygtuką pasirinkta detalė pridedama į pardavimo sąrašą (47 pav.). Atsidariusiame pardavimo sąraše randame tuos pačius mygtukus *Parduoti* ir *Parduoti ir spausdinti sąskaitą*, jų atliekami veiksmai jau aprašyti ankstesniame skyriuje.

Starteris
Opel Vectra C 2002-2008 2.2 DTI (125 Hp)
25 Eur

ABS kompiuteris
Opel Vectra C 2002-2008 2.2 DTI (125 Hp)
35 Eur

Parduoti **Parduoti ir spausdinti sąskaitą**

49 pav. Detalių pardavimo sąrašas.

Pristatyti – mygtukas naudojamas kada klientas pageidauja pristatyti detales nurodytu adresu. Paspaudus mygtuką atsidaro detalės pristatymo langas (49 pav.) kuriame reikia užpildyti su pristatymu susijusią informaciją arba pasirinkti iš esamos, jei klientas jau įvestas į duomenų bazę. Užpildžius formą paspaudžiamas mygtukas *Pristatyti* po šio veiksmo būna suformuojama ir atspausdinama sąskaita faktūra.

Detalės pristatymas ×

Pristatoma
Abs Kompiuteris

Pastovus klientas (pasirinkite įmonės kodą)
Pasirinkite klientą ▼

| | |
|----------------------------|--------------------------|
| Užsakovas(vardas, pavardė) | El .paštas |
| <input type="text"/> | <input type="text"/> |
| Tel. Nr. | Adresas |
| <input type="text"/> | <input type="text"/> |
| Pašto kodas | Pristatymo būdas |
| <input type="text"/> | Pasirinkite pristatymą ▼ |

Pristatyti

50 pav. Detalės pristatymo langas.

Ištrinti – paspaudus mygtuką detalė panaikinama iš detalių paruoštų pardavimui sąrašo, naudotojas apie tai informuojamas pranešimu.

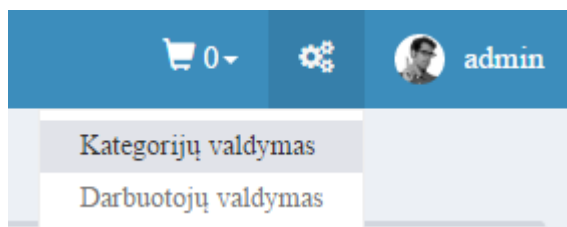
Spausdinti etiketę – atidaromas etiketės spausdinimo langas (50 pav.).

Atgal – mygtukas grąžina naudotoją į pardavimui paruoštų detalių sąrašą.

8.1.1. Kategorijų valdymas (detalių medžio sudarymas)

Detalių kategorijų medžio sudarymas yra pasiekiamas tik administratoriaus teisiams turintiems sistemos naudotojams.



















Kategorijų medžio valdymas pasiekiamas iš pagrindinio meniu pasirinkus: *Kategorijų valdymas*



51 pav. Kategorijų valdymo lango atidarymas.

Naujos kategorijos pridėjimas – naujos kategorijos pridėjimui iš atsidariusio kategorijų valdymo lango (45 pav.) pasirenkame *Pridėti kategoriją*, atsidariusiame lange (51 pav.) kategorijos pridėjimas pasirenkame kategorijos lygį, įrašome pavadinimą, nurodome kategorijos statusą ir

paspaudžiame mygtuką *Pridėti*. Po šių veiksmų kategorija bus pridėti ir sistemos naudotojas informuotas pranešimu.

| Kategorijų valdymas | | | |
|---|--|----------|---|
| Kategorijos | | Statusas | Veiksmas |
| | | | Pridėti kategoriją |
| Visas Automobilis | | Aktyvi |   |
| Elektronikos Detalės | | Aktyvi |   |
| Rankinio stabdžio kompiuteris | | Aktyvi |   |
| Zibinto aukscio reguliatorius (korektorius) | | Aktyvi |   |
| Navigacijos cd skaitytuvas | | Aktyvi |   |
| Liuko varikliukas | | Aktyvi |   |
| Variklio kompiuteris | | Aktyvi |   |
| Automagnetola | | Aktyvi |   |
| ESP greitejimo sensorius | | Aktyvi |   |

52 pav. Kategorijų valdymo langas.

Pridėti kategoriją ×

Kategorijos Lygis*

-- Pagrindinė --

Kategorijos pavadinimas*


Kategorijos statusas*

Aktyvi

Neaktyvi

Kategorijos aprašymas

53 pav. Naujos kategorijos pridėjimo langas.

Esamos kategorijos duomenų keitimas – kategorijos duomenims ar lygiui pakeisti naudojamas mygtukas  kuris atveria kategorijos redagavimo langą (52 pav.), šiame lange forma būna užpildyta pasirinktos kategorijos duomenimis kuriuos galima pakeisti, pakeitus norimus duomenis paspaudžiamas mygtukas *Atnaujinti*.

Pridėti kategoriją
×

Kategorijos Lygis*

Elektronikos Detalės ▼

Kategorijos pavadinimas*

Rankinio stabdžio kompiuteris

Kategorijos statusas*

Aktyvi
 Neaktyvi


Kategorijos aprašymas

nera

Uždaryti

Atnaujinti

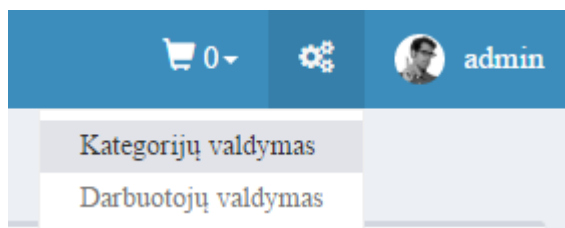
54 pav. Kategorijos redagavimo langas.

Kategorijos pašalinimas – kategorijos pašalinimui naudojamas mygtukas . Paspaudus mygtuką ties pasirinkta kategorija ši bus pašalinta iš sistemos, jei pasirinkta pašalinti kategorija turi sub kategorijų šios taip pat bus pašalintos kartu su pagrindine kategorija.

11.1.9. Darbuotojų valdymas

Darbuotojų valdymas yra pasiekiamas tik administratoriaus teisiams turintiems sistemos naudotojams.

Darbuotojų valdymas pasiekiamas iš pagrindinio meniu pasirinkus: *Darbuotojų valdymas*



55 pav. Darbuotojų valdymas.

Darbuotojo pareigų pridėjimas – darbuotojų pareigos pridedamos paspaudus mygtuką *Pareigos* ir atsidariusiame lange paspaudus *Naujos pareigos*. Atsivėrusioje formoje užpildoma pareigų pavadinimas, pasirenkamas lygis (**1** – Pilnas priėjimas prie visų sistemos funkcijų, **2** – pilnas automobilių ir detalių valdymas su paieškos galimybe, **3** – tik paieškos funkcija ir detalių pardavimas) ir paspaudžiamas mygtukas *Išsaugoti*. Po šių veiksmų pareigos išsaugojamos sistemoje.

Pareigų pavadinimas

Lygis

Pareigų aprašymas


Išsaugoti

56 pav. Naujų pareigybių pridėjimas.

Naujo darbuotojo pridėjimas – darbuotojas pridedamas paspaudus mygtuką *Naujas darbuotojas*. Atsidariusioje formoje (56 pav.) užpildoma naujo darbuotojo informacija, priskiriamos pareigos, ir paspaudžiamas mygtukas *Registruoti*.

| | | |
|---|--|---|
| <p>Darbuotojo nuotrauka</p>  | <p>Vardas, pavardė</p> <p>Vardas <input type="text"/></p> <p>Pavardė <input type="text"/></p> | <p>Pareigos</p> <p>Pasirinkite <input type="text"/></p> |
| | <p>Asmens kodas</p> <p>Asmens kodas <input type="text"/></p> | <p>Telefono numeris</p> <p>Telefono numeris <input type="text"/></p> |
| | <p>Adresas</p> <p>Namo nr., gatvė, miestas <input type="text"/></p> | <p>Elektroninis paštas</p> <p>E-paštas <input type="text"/></p> |
| | <p>Gimimo data</p> <p><input type="text"/> 1999-09-09</p> | <p>Slaptažodis <input type="text"/></p> <p>Pakartot slaptažodį <input type="text"/></p> |
| | <p>Registruoti</p> | |

57 pav. Naujo darbuotojo pridėjimo forma.

Darbuotojo duomenų redagavimas vykdomas paspaudus mygtuką: 

Darbuotojo duomenų ištrynimasis iš sistemos vykdomas paspaudus mygtuką: 

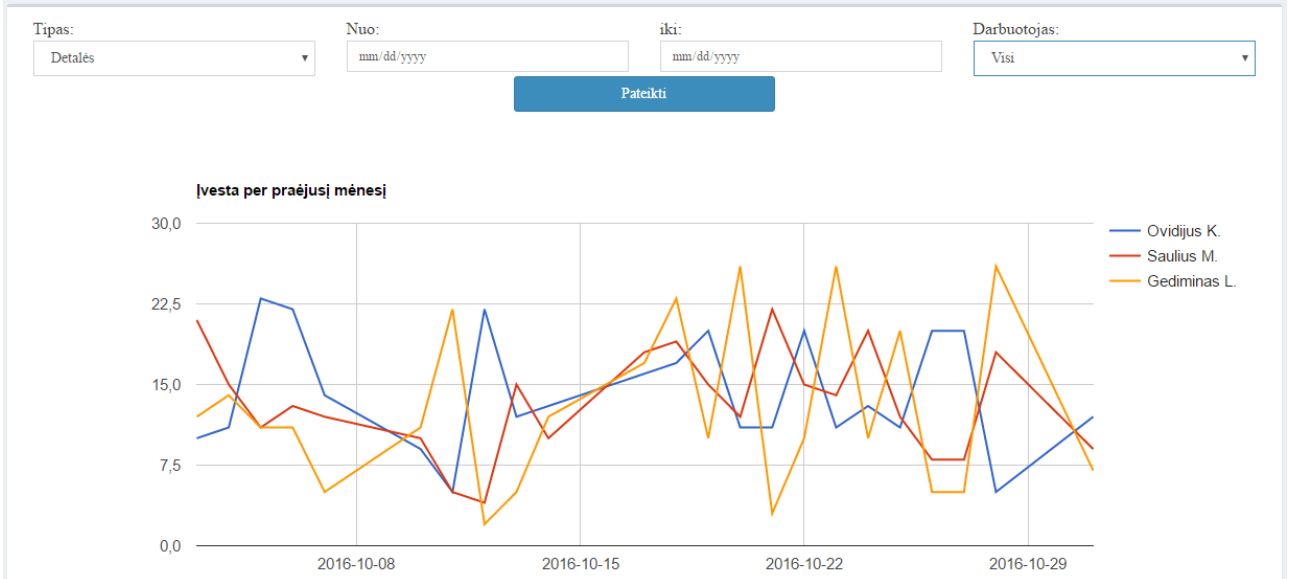
11.1.10. Ataskaitos

Ataskaitos pasiekiamos pagrindiniame meniu pasirinkus Ataskaitos, galima rinktis iš dviejų ataskaitų tipų *Įvedimų* (pateikiama kiek ir kokių detalių ar automobilių įvedė darbuotojas) ir *Pardavimų* (Kiek ir kokių detalių buvo parduota per pasirinktą laikotarpį).

Ataskaitos formavimas:

Paspaudus mygtuką *Ataskaitos* -> *Įvedimų* pateikiama praėjusio mėnesio įvedimų ataskaita (57 pav.)

Įvedimų ataskaitos



58 pav. Įvedimų ataskaita.

Norint gauti ataskaitą pagal pasirinktus parametrus galima juos nurodyti viršuje grafiko esančioje parametrų juostoje (58 pav.), pasirinkus parametrus ir paspaudus mygtuką pateikti, ataskaita bus atnaujinta pagal nustatytus duomenis.

Tipas: Nuo: iki: Darbuotojas:

Pateikti

59 pav. Ataskaitos parametrų juosta.

11.1.11. Filtrų taikymas

Filtrai sistemoje pasirenkami paspaudus mygtuką *Taikyti filtrą* ir atsidariusiuose filtro nustatymuose pasirinkus norimus duomenis paspaudžiamas mygtukas *Atrinkti*. Filtro duomenys išvalomi paspaudus mygtuką *Anuliuoti*.

Užsakymai

| Užsakymo nr. | Užsakymo data | Statusas | Užsakovas | Kontaktai | Veiksmai |
|--------------------------------------|--|----------|-------------------------|-------------------------------|---|
| <input type="text"/> | Nuo: <input type="text"/> Iki: <input type="text"/> | Vykdomas | | | <input type="button" value="Atrinkti"/> <input type="button" value="Anuliuoti"/> |
| 2016-11-27 16:20:462HGES16511H595741 | 2016-11-27 16:22:06 | Vykdomas | tomas, pavardenis | 8688811111, tomas@mail.com | <input type="button" value="Atrinkti"/> <input type="button" value="Anuliuoti"/> |
| 2016-11-25 08:38:402HGES16511H595741 | 2016-11-25 08:39:08 | Vykdomas | 302489137, Uab "Lexita" | +370-6528899, lexita@mail.com | <input type="button" value="Atrinkti"/> <input type="button" value="Anuliuoti"/> |
| 2016-11-24 10:38:09 | 2016-11-24 10:38:09 | Vykdomas | ovidijus1, pavardenis | 86800000, ovizz@admin.com | <input type="button" value="Atrinkti"/> <input type="button" value="Anuliuoti"/> |

60 pav. Filtro taikymas užsakymų lentelės duomenims atrinkti.

11.2. 2 PRIEDAS – TERMINŲ IR SANTRUMPŲ ŽODYNAS

- AJAX** – Asynchronous JavaScript and XML (asinchroninis JavaScript ir XML programavimas)
- API** – Application Programming Interface (aplikacijų programavimo sąsaja)
- CMS** – Content Management System (turinio valdymo sistema)
- CSS** – Cascading Style Sheets (stiliaus kalba, skirta nusakyti kita struktūrine kalba aprašytam dokumentui vaizduoti)
- DB** – Data base (duomenų bazė)
- DFD** – data flow diagram (duomenų srautų diagrama)
- DBMS** – Database Management System (duomenų bazių valdymo sistema)
- GUI** – Graphical User Interface (grafinė vartotojo sąsaja)
- HTML** – HyperText Markup Language (hiperteksto žymėjimo kalba)
- HTTPS** – Hypertext Transfer Protocol Secure (saugus hiperteksto perkėlimo protokolas)
- MVC** – Model-View-Controller (programinės įrangos architektūra)
- OOP** – Object-Oriented Programming (objektiškai orientuotas programavimas) XML - eXtensible Markup Language (ištęsiama žymėjimo kalba)
- ORM** – Object-relational mapping (virtuali objektų duomenų bazė)
- OS** – Operating system (Operacinė sistema)
- PI** – programinė įranga
- QR** – Quick Response (dvimatis brūkšninis kodas)
- RUP** – Rational unified process (rational unifikuotas procesas)
- SOC** – Separation Of Concerns (interesų atskirtis)
- URL** – Uniform Resource Locator (globalus išteklių lokatorius)
- UPC** – Universal Product Code (universalus kodas produktams žymėti)
- XSS** – Cross site scripting (IT sistemų pažeidžiamumas, dažniausiai aptinkamas tinklalapiuose, kuris leidžia įterpti papildomą programinį kodą)