

## Duomenų bazės prototipo (bandomosios versijos) sukūrimas

### Donatas Daugirdas

*Šiaulių valstybinė kolegija, Informatikos mokslų katedros lektorius*

*Šiaulių valstybinė kolegija / Šiauliai State Higher Education Institution, Lithuania; Lecturer of the Department of Informatics Science*

d.daugirdas@svako.lt

### Justinas Zatorskis

*Šiaulių valstybinė kolegija, Programų sistemų studijų programos studentas*

*Šiaulių valstybinė kolegija / Šiauliai State Higher Education Institution, Lithuania; Student of the Software engineering study program*

justinas.zatorskis@stud.svako.lt

### Anotacija

Straipsnyje analizuojamas duomenų bazės prototipo kūrimas įmonės užsakymu. Atliktoje panašių sistemų analizėje buvo palygintos 3 panašios sistemos. Lyginant galimas kūrimo technologijas buvo atsižvelgiama į reikalavimus kuriamai sistemai, sukurtas veikiantis prototipas įmonės naudojamoje operacinėje sistemoje. Suprojektuotos diagramos, iliustruojančios sistemos veikimo principus, vartotojų panaudos atvejus ir duomenų bazės struktūrą. Nustatyti funkcionalumai sistemos naudojimui. Suprogramuota aplikacija, sistemos prototipas „STRWarehouseApp“ kuris duomenis saugo ir apdoroja „MySQL“ duomenų bazėje. Sistemos prototipo testavimas buvo atliekamas viso programavimo proceso metu.

**Reikšminiai žodžiai:** duomenų bazė, prototipas, sandėlio inventoriaus valdymo sistema, projektavimas.

## Creation of the database prototype

### Summary

The article analyzes the creation of a database prototype commissioned by the company. A similarity analysis compared 3 similar systems. When comparing the possible development technologies, the requirements for the system being developed were taken into account, and a working prototype was created in the operating system used by the company. Designed diagrams illustrating the principles of system operation, user use cases and database structure. Functionality for system use was defined. Programmed application, system prototype "STRWarehouseApp" which stores and processes data in MySQL database. System prototype testing was performed throughout the programming process.

**Keywords:** database, prototype, warehouse inventory management system, design.

### Įvadas

*Temos aktualumas.* Kiekvienai įmonei ar organizacijai, kuri turi kokį nors sandėliuojamą inventorių, reikia turėti ir to inventoriaus sąrašą. Sandėlio duomenų bazės yra būtinos įmonėms, kad jos galėtų saugiai ir efektyviai valdyti savo sandėlių atsargas. Duomenų bazės leidžia įmonėms laikyti informaciją apie sandėlio inventorių, prekių gabenimo ir priėmimo laikus, prekių porcijų kiekius, taip pat galimybes, sandėlio darbuotojų vardus ir kita informacija, reikalinga valdyti sandėlių operacijas. Be to, gerai sukurta sandėlio duomenų bazė gali padėti įmonėms sumažinti klaidas ir išlaidas, kurios kyla dėl netikslių ar neteisingų duomenų ar procesų.

Jeigu organizacijos sandėlio sąrašas yra 500 daiktų, iš kurių kiekvienas gali turėti po 50 įrašų, tai tokius duomenis tampa problematiška apdoroti be papildomų sistemų. Šiam tikslui gali būti panaudotos inovatyvios įmonės sandėlio valdymo sistemos. Tokios sistemos padeda sekti ir valdyti įmonės sandėliavimo procesus, trumpalaikį ir ilgilaikį turtą ar net įmonės veiklos pelningumą arba nuostolingumą bei kitą informaciją, susijusią su verslo subjekto vykdomais procesais realiu laiku, tai padeda išvengti galimų finansinių nuostolių atsirandant klaidoms sandėlio darbo procese. Jų integracija ir tobulinimas neatsiejamas nuo augančios įmonės veiklos apimtys. Šios sistemos padeda sekti sandėlio inventoriaus priežiūrą, likutį, techninio aptarnavimo datas, tai sumažina riziką netekti

užsakymų dėl prekių sandėlyje trūkumo ir pagerina įmonės užsakymo įvykdymo laiką. Jei apskaita neparodo tikslios sandėlio turto būklės ir jo judėjimo, tai gali lemti nepakankamą priežiūrą ir nepakankamą turto tvarkymą. Tai gali prisidėti prie turto praradimo, nuvertėjimo arba jo trūkumas gali sutrikdyti užsakymų vykdymą, o perteklius gali lemti perteklinį sandėlio užimtumą ir nereikalingas išlaidas.

Nepakankamas verslo apskaitos efektyvumas gali lemti netikslumus skaičiuojant sandėlio turto kieki, vertę arba kitus susijusius rodiklius. Tai gali sukelti netikslumus planuojant sandėlio poreikius, užsakymus ir tiekimo grandinės valdymą. Sukurta duomenų bazė gali būti puiki priemonė pagerinti verslo veiklą. Kai duomenys yra įvesti į duomenų bazę, galima pradėti naudoti įvairius metodus ir užklausas, kad būtų išgauta informacija iš duomenų bazės. Tai gali apimti duomenų analizę, ataskaitų generavimą, tendencijų stebėjimą ir kitas veiksmingas priemones, kurios padės priimti geriausius verslo sprendimus.

*Tyrimo problemą* galima suformuluoti tokiais probleminiais klausimais: Pagal kokias jau veikiančias panašias sistemas galima sukurti sandėlio valdymo sistemos prototipą? Kuri technologija yra tinkamiausia tokio prototipo sukūrimui? Kokie reikalavimai keliami kuriamai sistemai? Kaip suprogramuoti sandėlio valdymo sistemos prototipą?

*Tyrimo objektas:* duomenų bazės prototipas.

*Tyrimo tikslas* – sukurti ir įdiegti sandėlio valdymo sistemos prototipą į „Windows“ OS palaikantį įrenginį.

*Tyrimo uždaviniai:* 1) Išanalizuoti įmonę, jos poreikius sistemai sukurti; 2) Išanalizuoti panašias sistemas; 3) Palyginti galimas kūrimo technologijas, išsirinkti tinkamiausią; 4) Nustatyti reikalavimus, keliamus sistemai kaip vienetui; 5) Suprogramuoti sistemos prototipą-aplikaciją „Windows“ operacinėms sistemoms; 6) Atlikti sistemos prototipo testavimą.

*Tyrimo metodai:* interviu su įmonės vadovu, mokslinės literatūros šaltinių analizė, panašių veikiančių sistemų analizė, projektavimas, kuriamo individualaus projekto testavimas.

## **Įmonės poreikių analizė**

UAB „Šiaulių tauro ryšiai“ yra įmonė, užsiimanti elektrotechnikos ranga. Kaip teigiama įmonės internetinėje svetainėje [8], įmonė specializuojasi įvairiuose elektrotechnikos rangos darbuose, tokiuose kaip saulės kolektorių, apsaugos sistemų, telekomunikacijų ir kiti montavimo darbai.



**1 pav.** Įmonės UAB „Šiaulių tauro ryšiai“ logotipas [8].

Įmonėje naudojama programinė įranga:

- „Windows 10 PRO“ operacinė sistema,
- „MS Office“ paketas,
- „DSM 6“ operacinė sistema.

Interviu su įmonės vadovu metu buvo išsiaiškinta, kad įmonei reikalinga sandėlio apskaitos sistema. Įmonė iki sistemos įdiegimo sandėlį valdė tik „MS Excel“ įrankiu, dėl ko nebuvo galima įvesti patvirtinimų apie prekių išdavimą, gavimą. Norint peržiūrėti sandėlio inventorių, šia galimybę turėjo tik sandėlininkas, o vadovui norint pasitikrinti bendrą situaciją bei kontroliuoti sandėlio darbą reikėjo rankiniu būdu persiųsti failą. Toks variantas nėra tinkamas, kadangi faile nebuvo jokios kontrolės nei istorijos, kas ir kada vykdė pakeitimus. Sandėlio vadovas išreiškė susirūpinimą dėl prekių grąžinimo kontrolės, įmonėje įsisenėjusi problema, jog darbuotojai nepraneša apie nepanaudotų prekių grąžinimą, o tiesiog palieka nepanaudotas prekes sandėlyje be konkretaus likučių kiekio. Šiai problemai išspręsti įmonės vadovui buvo pasiūlytas funkcionalumas dėl prekių grąžinimo ir grąžinimo patvirtinimo. Darbų vadovas, grąžindamas prekes į sandėlį, turi užpildyti formą su

kiekvienos gražinamos prekės kiekiu bei kodu. Po formos užpildymo sandėlininkas, prisijungęs prie savo paskyros, matys laukiantį nepatvirtintą gražinimą ir pasitikrinęs, jog duomenys teisingi, patvirtins gražinimą. Analogiška situacija vykdoma ir su prekių užsakymu bei išdavimu: darbų vadovas pateikia sąmatą sandėliui, kas jam reikalinga konkrečiam objektui, tada sandėlininkas mato šią informaciją ir patitkrina sandėlio likutį. Jeigu prekių trūksta, jos užsakinėjamos pagal poreikį. Prekės gali būti išduodamos ne visos vienu metu, kadangi objekto įgyvendinimo laikas gali skirtis, todėl užsakymas uždaromas tik pasibaigus projektui. Užsakymo lange matoma informacija apie sąmatą bei apie išduotas ir dar neišduotas prekes. Kadangi įmonės darbuotojams darbo metu gali prireikti papildomų prekių, sistemoje turi būti galimybė papildyti užsakymą. Įmonės vadovas taip pat pateikė reikalavimą dėl besibaigiančio likučio paryškkinimo, buvo pasiūlyta siųsti perspėjimą apie besibaigiantį likutį elektroniniu laišku. Su įmonės darbuotojais bendrauta ne kartą, tiek fiziškai nuvykus, tiek telefonu.

### **Analogų analizė**

Analogų analizei parinktos analogiškos ERP sistemos: „Oracle NetSuite ERP“, „Microsoft Dynamics 365 Business Central“, „StrategicERP“. ERP (Enterprise Resource Planning) sistema yra integruota verslo valdymo programa, skirta efektyviai valdyti ir koordinuoti įmonės skirtingas veiklos sritis. Šios sistemos apima įvairius modulius, kurie atitinka skirtingas verslo funkcijas ir procesus, ir sujungia juos į vieną centralizuotą informacijos sistemą [2]. Jos pasirinktos dėl populiarumo tarp vartotojų bei dėl žinomų ir didelių kompanijų, kurios kuria ir vysto šias konkrečias sistemas.

Toliau buvo analizuojamos konkrečios analogiškos sistemos ir sudaromi jų vertinimai pagal kriterijus, kurie siūlomi David Luther straipsnyje [1]. Vertinimo kriterijai:

1. *Verslo poreikis:* Nusakoma, kiek sistema atitinka verslo poreikius, kiek tai aktualu verslo subjektams. Atsakoma į klausimus: Ar ERP sistema atitinka verslo poreikius? Ar ERP sistema reikalinga konkrečiame verslo segmente? Vertinimas: „+“ ar „-“, kai „-“ – netenkinami verslo poreikiai, kai „+“ – tinkamai išpildyti verslo poreikiai sistemai.
2. *Sistemos funkcionalumas:* Nusakoma, kiek sistema funkcionali ir kokios jos galimybės. Atsakoma į klausimus: Ar sistemos vartotojo sąsaja pakankamai intuityvi ir paprastai perprantama? Ar sistema galima naudotis iš mobiliojo įrenginio? Vertinimas: „+“ ar „-“, kai „-“ – sistemos funkcionalumas prastas, sistema veikia su trukdžiais, sistema nepasiekama iš mobiliojo įrenginio arba sudėtinga vartotojo sąsaja; kai „+“ – sistema pasiekama iš mobiliojo įrenginio, funkcionalumas veikia be trukdžių, vartotojo sąsaja pakankamai paprasta ir greitai perprantama.
3. *Bendra sistemos kaina:* Nusakoma bendra sistemos kaina verslo subjektui. ERP sistemų pardavinėjimo modelis – kaina vartotojui už licenciją. Vertinimas: „+“ ar „-“, kai „-“ – sistemos kaina vartotojui daugiau nei 20 Eur/mėn., kai „+“ – sistemos kaina vartotojui mažiau nei 20 Eur/mėn.
4. *Sistemos integralumas:* Nusakoma sistemos galimybė integruotis į jau naudojamas verslo sistemas. Vertinimas: „+“ ar „-“, kai „-“ – sistema neturi galimybių integruotis su kitomis sistemomis, kai „+“ – sistema turi galimybes integruotis su kitomis sistemomis.
5. *Pardavėjo pagalba:* Nusakoma, ar sistemos tiekėjas suteikia visą galimą pagalbą vartotojui, susijusią su sistemos eksploatavimu. Vertinimas: „+“ ar „-“, kai „-“ – sistemos tiekėjas nesuteikia pagalbos, tik pačią sistemą, kai „+“ – sistemos tiekėjas suteikia pagalbą, susijusią su sistemos eksploatavimu.
6. *Naudojama technologija:* Nusakoma, ar sistemos naudojama technologija yra pakankamai lanksti, jog būtų galima pridėti papildomas funkcijas verslo subjekto veiklai vystyti. Vertinimas: „+“ ar „-“, kai „-“ – naudojama technologija nėra pakankamai gera, kad būtų įvedami naujiniai, arba technologija pasenusi ir ji nebėra vystoma iš techninės pusės, kai „+“ – naudojama technologija yra lanksti ir nuolat vystoma iš techninės pusės.
7. *Tiekėjo komunikacija:* Nusakoma tiekėjo komunikacija su vartotojais, ar tiekėjas informuoja apie būsimus atnaujinimus, papildomo funkcionalumo diegimą, sistemos migraciją į kitas platformas

ir pan. Vertinimas: „+“ ar „-“, kai „-“ – tiekėjas nepraneša informacijos apie sistemos pasikeitimus, kai „+“ – tiekėjas suteikia visą aktualią informaciją apie sistemos pasikeitimus vartotojui.

8. *Sistemos prisitaikymas prie vartotojo*: Nusakoma galimybė kiekvienam vartotojui keisti vartotojo sąsają ar funkcionalumą pagal poreikius. Vertinimas: „+“ ar „-“, kai „-“ – redagavimo galimybės nėra, sistema paroduodama tokia, kokią ją sukūrė tiekėjas, ir vartotojas negali jos redaguoti pagal savo poreikius, kai „+“ – sistemą galima redaguoti iš vartotojo pusės pagal jo poreikius.

„Oracle NetSuite ERP“ yra didelių galimybių ERP sistema, kuri grįsta debesų kompiuterija. Sistema padeda verslo subjektams valdyti procesus efektyviau su realaus laiko duomenų perdavimu ir rodymu vartotojams bei pateikia finansinio pelningumo statistiką. Sistema apima finansų, sandėlio valdymo, inventoriaus valdymo ir tiekimo grandinės procesus [5].

„Microsoft Dynamics 365 Business Central“ sistema yra didelės apimties ERP sistema, teikianti vartotojui daug funkcionalumo, gerą techninę pagalbą ir integralumą. Sistema puikiai išvystyta ir paremta naujausiomis technologijomis [17]. Ji skirta mažiems ir vidutiniams verslo subjektams, veikia debesų kompiuterijos principu. Tačiau tai yra pakankamai sudėtinga sistema vartotojui ir reikalauja papildomų žinių, norint išnaudoti visas sistemos galimybes.

„Strategic ERP“ sistema apima įvairaus spektro verslo subjektų procesus. Ji gali pasiūlyti standartizuotus procesų valdymo sprendimus. Sistema grįsta debesų kompiuterija bei naudoja dirbtinio intelekto elementus [18]. Turint omenyje sistemos funkcionalumo apimtį, ji skirta didelėms įmonėms ir nėra gerai orientuota į mažus ar vidutinius verslus. Ji pakankamai sudėtinga, tačiau teikia pagalbą ir mokymus klientams. Komunikacija apie sistemos atnaujinimus taip pat išpildyta, klientai gauna informaciją apie būsimus sistemos pakeitimus.

Atsižvelgiant į vertinimo duomenis, matoma, jog analizuotos sistemos pritaikytos dideliems verslams ir tai veikia sistemos įsigijimo bei išlaikymo kaštus. Tačiau sistemos gerai išvystytos ir teikia daug funkcionalumo verslo subjektams procesų valdymo srityje.

#### 1 lentelė. Analogiškų ERP sistemų vertinimas

Vertinama sistema	„Oracle NetSuite ERP“	„Microsoft Dynamics 365 Business Central“	„Strategic ERP“	Kuriama sistema
<b>Vertinimo kriterijus</b>	<b>Vertinimas</b>			
Verslo poreikis	+	+	+	+
Sistemos funkcionalumas	+	+	+	+
Bendra sistemos kaina	-	-	-	+
Sistemos integralumas	-	+	-	+
Pardavėjo pagalba	+	+	+	+
Naudojama technologija	+	+	+	+
Tiekėjo komunikacija	+	+	+	+
Sistemos prisitaikymas prie individualaus vartotojo poreikių	-	-	+	+

Įvertinus analogiškas sistemas, bendrą ERP sistemų pasiūlą ir užsakovo reikalavimus, nutarta kurti mažos įmonės sandėlio procesų valdymo poreikius atitinkančią duomenų bazės prototipą (toliau – sistema), kuri nereikalautų didelių investicijų, naudotų atviro kodo bibliotekas, jos funkcionalumas būtų praktiškai pritaikomas inventoriaus bei sąnaudų valdyme.

#### ERP sistemų apžvalga

ERP angl. „Enterprise Resource Planning“ – tai tokios sistemos, kurios padeda verslo subjektui valdyti rutininis procesus [2], tokius, kaip darbuotojų ar inventoriaus apskaita, rizikų ir pelningumo analizės, sandėlio likučių peržiūra ir tiesioginiai integruoti užsakymai iš sistemos ar įmonės inventoriaus priežiūra ir rutininiai patikrinimai, sąnaudų sekimas.

Kaip teigiama oficialioje „Oracle“ svetainėje [2], ERP sistemos integruojamos į kitas verslo subjekto naudojamas sistemas, taip susiedamos visus įmonės vykdomus procesus. Tai pasiekama vykdam duomenų perdavimą tarp įmonės naudojamų sistemų.

ERP sistemų galimybės priklauso nuo verslo subjekto poreikių. Dažnu atveju ERP sistema yra pritaikoma konkrečiam verslo subjektui, tačiau yra sistemų, kurios pardavinėjamos be pritaikomumo, kaip standartinės versijos [7]. Tokia stacionari ir ne lanksti sistema dažnu atveju neatitinka visų verslo subjekto poreikių, todėl kiekvienai įmonei didelis privalumas – sistemos integracija ir pritaikomumas verslo poreikiams ir konkrečioms verslo subjekto procesams.

Trumpai aptarus verslo programinę įrangą, reikia paminėti, kad tokios sistemos sudėtingos ir reikalauja individualaus diegimo bei integracijos kiekvienoje įmonėje. Todėl įmonė, norėdama pasinaudoti sistemos teikiamais privalumais, turi nuolat investuoti į sistemos funkcionalumo atnaujinimus. Be to, verslo procesai nuolat kinta ir tobulėja, todėl ir naudojama sistema turi būti pritaikyta pagal procesų pasikeitimus. Neinvestuojant į sistemos atnaujinimus, ji gali tapti net žalinga, jei neatnaujinama su procesais realiu laiku, todėl „Agile“ principo taikymas šioms sistemoms yra ypatingai svarbus.

### Programinės įrangos analizė ir pasirinkimas kuriamai sistemai

„Microsoft Windows“ operacinėms sistemoms pritaikytoms programoms kurti yra platus įrankių, programavimo kalbų ir platformų pasirinkimas [3]. Kadangi iš „Windows“ OS pasiekiamos tiek „WEB“ aplikacijos, tiek „executable“ failai, tiek „jar“ failai, šioje vietoje galima rinktis iš PHP, JAVA, C# ir kitų kalbų. Toliau lentelėse pateikti pagrindiniai paminėtų kalbų trūkumai ir privalumai.

1 lentelė. PHP programavimo kalbos privalumai [12]

PHP privalumai	
1.	PHP yra atviro kodo nemokama programavimo kalba.
2.	Nepriklausanti konkrečiai platformai kalba, kuri gali veikti UNIX, Linux, Windows ir kt. operacinėse sistemose.
3.	Sąlyginai lengvai įsisavinama, aukšto lygio programavimo kalba.
4.	Pastovus naujinių palaikymas.
5.	PHP aplikacijos išgauna sąlyginai didelį procesų veikimo greitį net ir su lėtesniu interneto greičiu, lyginant su kitomis programavimo kalbomis.
6.	PHP suprojektuota taip, kad kuo labiau palengvintų programuotojui apdoroti rašomą kodą.
7.	Lengvas parašyto kodo pernaudojimas, padeda išvengti perteklinio kodo dubliavimo.
8.	PHP turi integruotą duomenų bazės prisijungimo modulį, kuris supaprastina aplikacijos surišimą su duomenų baze.
9.	Didelė bendruomenė, dėl ko lengva rasti informacijos apie reikalingus algoritmus bei bibliotekų pasirinkimas.

2 lentelė. PHP programavimo kalbos trūkumai [12]

PHP trūkumai	
1.	Nėra pakankamai saugi dėl atviro kodo ir dėl ASCII tekstinių failų lengvo pasiekiamumo.
2.	Nėra pritaikyta didelėms sistemoms.
3.	PHP karkasai turi naudoti integruotus funkcionalumus tam, kad išvengtų papildomo kodo rašymo.
4.	Kuo daugiau naudojama karkasų ir integruotų funkcijų, tuo labiau silpnėja internetinių sistemų greitis.
5.	PHP negalima keisti esminio veikimo.
6.	PHP karkasai nėra lygiaverčiai su kitų programavimo kalbų karkasais funkcionalumo prasme.
7.	PHP neturi pakankamai gerų klaidų paieškos (DEBUG) įrankių, dėl ko paliekama daugiau klaidų, o esamas klaidas žymiai sunkiau aptikti.
8.	Imituojami „JAVA“ kalbos fragmentai, kurie komplikuoja tam tikrų funkcijų ir kodo fragmentų naudojimą.

Pagal detalizuotus PHP programavimo kalbos trūkumus ir privalumus daroma išvada, kad PHP yra aukšto lygio programavimo kalba, tai yra pakankamai geras pasirinkimas „Web“ aplikacijų kūrimui, tačiau turi tam tikrų spragų dėl saugumo, klaidų paieškos ir funkcionalumo. Taip pat paminėtinas „multithreading“ PHP kalboje trūkumas [12], jis pasiekiamas naudojant papildomas bibliotekas, tačiau jos dažnai turi rimtų defektų, kurie gali sukelti smarkių padarinių sistemai ir jos saugumui, o jų klaidų paieška labai problematiška ir užima daug laiko bei reikalauja išteklių.

### 3 lentelė. JAVA programavimo kalbos privalumai [13]

JAVA privalumai	
1.	Objektinio programavimo kalba.
2.	JAVA yra saugi programavimo kalba dėl savo operatyviosios atminties naudojimo principų.
3.	Pigi ir ekonomiška programavimo kalba.
4.	Nepriklausanti konkrečiai platformai kalba, kuri gali veikti MAC, Linux, Windows ir kt. operacinėse sistemose.
5.	JAVA yra aukšto lygio programavimo kalba.
6.	Automatiniai „šiukšlių surinkėjai“.
7.	JAVA programavimo kalba palaiko „multithreading“ (kelių procesų vienu metu veikimą atskiruose linijose).
8.	Stabili kalba lyginant su alternatyvomis.

### 4 lentelė. JAVA programavimo kalbos trūkumai [13]

JAVA trūkumai	
1.	JAVA kodas yra lėtas ir naudoja daug kompiuterio resursų.
2.	JAVA GUI nėra patrauklus lyginant su alternatyvomis.
3.	Nėra atsarginės kopijos funkcionalumo.
4.	JAVA naudoja, palyginus su kitomis kalbomis, daug operatyviosios atminties, pvz., lyginant su alternatyvomis C ir C++. Vykstant „šiukšlių surinkimui“ galimi sistemos trukdžiai.
5.	Komplikuota sintaksė.

Pagal pateiktus JAVA programavimo kalbos trūkumus ir pranašumus daroma išvada, kad JAVA yra aukšto lygio programavimo kalba, kaip ir PHP, tačiau ji stabilesnė už PHP programavimo kalbą [14]. Ji turi stabilų „multithreading“ mechanizmą, tačiau naudoja gerokai didesnius resursų kiekius. Kaip teigia Nicholas Rini, JAVA programavimo kalba turi gerokai geresnę klaidų paieškos sistemą lyginant su PHP [14], dėl to programuojant su JAVA programavimo kalba galima išvengti daug nesklandumų tiek programavimo, tiek eksploatavimo procesuose.

### 5 lentelė. C# programavimo kalbos privalumai [9]

C# privalumai	
1.	Objektinio programavimo kalba, tačiau yra ir procedūrinio programavimo pasiekiamumas naudojant C++/CLI projekto struktūrą.
2.	Automatinis tipų priskyrimas kintamiesiems.
3.	Automatinis „šiukšlių surinkėjas“.
4.	Nepriklausanti konkrečiai platformai kalba, kuri gali veikti MAC, Linux, Windows ir kt. operacinėse sistemose.
5.	Integruota versijavimo kontrolė.
6.	Integruotas GUI redaktorius.
7.	Integruotas „NuGet“ bibliotekų įrankis.
8.	Išsamiai aprašyta dokumentacija.
9.	Optimizuotas resursų naudojimas.
10.	„Legacy“ (nebevystomų karkasų) palaikymas.
11.	„multithreading“ funkcionalumas.
12.	Suderinamumas su žemo lygio programavimo kalbomis, tokiomis kaip C, C++.
13.	Nėra „pointer“ tipų.

### 6 lentelė. C# programavimo kalbos trūkumai [9]

C# trūkumai	
1.	Galimi atminties nutekėjimo atvejai.
2.	Didelis nebeatnaujinamų karkasų pasirinkimas.
3.	Sąlyginai sudėtinga programavimo kalba.
4.	Ne visada tikslūs klaidų pranešimai.
5.	Nėra „pointer“ tipų.

Pagal pateiktus C# programavimo kalbos privalumus ir trūkumus daroma išvada, jog C# yra pranašiausia iš trijų analizuotų programavimo kalbų. Tai yra aukšto lygio programavimo kalba, kuri turi suderinamumą su žemo lygio programavimo kalbomis. Būtent dėl šios priežasties su šia programavimo kalba galima pasiekti geresnių rezultatų, išnaudojant mažiau kompiuterio pajėgumų. Tačiau šis funkcionalumas turi ir trūkumų: kintamųjų tipai skiriasi ir dažnu atveju reikiant panaudoti neprimityvius kintamuosius, kyla problemų su jų perdavimu. Šios problemos išsprendžiamos, tačiau

gali užimti daug laiko tinkamai suprojektuoti sistemos metodus. Ji pranašesnė dėl labai išsamios oficialios dokumentacijos apie funkcijas, tačiau ji yra smarkiai priklausoma nuo karkasų, tad rekomenduotina naudoti ilgo palaikymo karkasus. Tačiau sistemos, kuriose panaudoti senesni karkasai, yra palaikomos „legacy“ principu, bet jie nebegauna saugumo atnaujinimų iš Microsoft [9].

Atsižvelgus į programavimo kalbų privalumus ir trūkumus bei užsakovo keliamus reikalavimus sistemai, priimta sistemos kūrimui naudoti C# programavimo kalbą. Kadangi pasirinkta programavimo kalba C#, sistemos kūrimui yra pasitelkti Microsoft įrankiai, prisijungimui prie duomenų bazės naudojamas „Bitvise“ įrankis, duomenų bazei kurti naudojamas „Entity“ karkasas.

Projektui kurti naudojama programinė įranga:

- C# programavimo kalba;
- Visual Studio 2022 Community;
- .NET 4.7.2;
- „Entity“ karkasas;
- MY SQL;
- „Bitvise SSH Client“.

### **„.NET“ karkaso, „NuGet“ paketų ir „Windows“ OS apžvalga**

Oficialioje „Microsoft“ svetainėje [4] nurodoma, kad .NET karkasas yra „Microsoft“ sukurtas programavimo karkasų rinkinys, skirtas kurti ir vykdyti „Windows“ operacinėje sistemoje veikiančias programas. Šis karkasas apima plačią bibliotekų ir įrankių rinkinį, kurie programuotojams leidžia kurti „Windows“ programų, Web aplikacijų, mobiliąsias aplikacijas ir daugelį kitų sprendimų.

.NET karkaso bibliotekų rinkinys yra labai plataus funkcionalumo ir teikia daug įvairių įrankių, kurie padeda programuotojams kurti aukštos kokybės aplikacijas „Windows“ operacinėje sistemoje. Ši biblioteka yra sudaryta iš daugybės klasės bibliotekų, kurios apima įvairias programavimo sritis, tokių kaip įvesties / išvesties operacijos, failų tvarkymas, tinklo ryšių valdymas, kriptografija, saugumas, XML tvarkymas ir daugelis kitų.

„NuGet“ yra .NET platformos atviro kodo programinės įrangos (angl. open-source software) valdymo sistema, skirta valdyti ir tvarkyti paketus (angl. packages), kurie yra paketų bibliotekos, skirtos atlikti konkrečius uždavinius. NuGet yra prieinama per Visual Studio ir leidžia programuotojams lengvai instaliuoti, atnaujinti ar šalinti bibliotekas ir jų versijas, kurios naudojamos „Windows“ operacinėje sistemoje.

Kuriant programą naudojant .NET karkasą, programuotojas gali naudoti tiek standartines .NET bibliotekas, tiek ir daugybę trečiųjų šalių bibliotekų, kurios yra prieinamos per NuGet. Šios bibliotekos gali būti atsisiunčiamos ir įdiegiamos automatiškai, jas gali valdyti ir tvarkyti NuGet. Tai padeda programuotojams daug greičiau ir efektyviau kurti aplikacijas, kadangi jie turi prieigą prie daugybės funkcijų ir įrankių, kurie jau yra sukurti ir optimizuoti, o taip pat gali naudoti ir pasidalinti atviro kodo bibliotekomis, skirtomis „Windows“ operacinei sistemai.

### **UAB „Šiaulių tauro ryšiai“ sandėlio inventoriaus apskaitos sistemos projektavimas ir realizavimas**

Šiame skyriuje pateikta parengta projekto sistemos specifikacija, panaudos atveju, sistemos sekų, duomenų bazės modeliai ir aprašomas jų veikimas bei sąveika. Projektas kurtas etapais: kuriama aplikacija, duomenų bazė, viskas sujungiama į vientisą projektą gaunant sistemos prototipą. Prototipas diegiamas įmonėje, po testavimo ir patvirtinimo tampa vientisa veikiančia sistema. Pateikiami programavimo proceso žingsniai ir apžvelgiamos problemos, su kuriomis buvo susidurta programavimo metu, bei kaip šios problemos buvo išspręstos.

Versijavimui naudota „GitHub“ laikmena, kurios pagalba buvo galima sekti pakeitimus, grąžinti panaikintus pakeitimus, pasiekti kodą iš bet kurio įrenginio, esant reikalui programuoti iš skirtingų kompiuterių. Tai yra ypatingai svarbus įrankis dirbant „Agile“ metodologijos principu, nes dažnai pasitaiko situacijų, kai užsakovas spontaniškai sugalvoja papildomą pakeitimą arba tenka grąžinti

panaikintą funkcionalumą. Su „GitHub“ laikmenos pagalba galima pasiekti kodą, kuris buvo prieš įdiegiant pakeitimus. Programavimo procesas vyko pasitelkiant „Agile“ metodologijos principą. Reikalingi pakeitimai, kurie buvo nenumatyti pradžioje, jie visi buvo įgyvendinami sistemos kūrimo eigoje.

## Programos specifikacija ir taikomos duomenų saugos priemonės

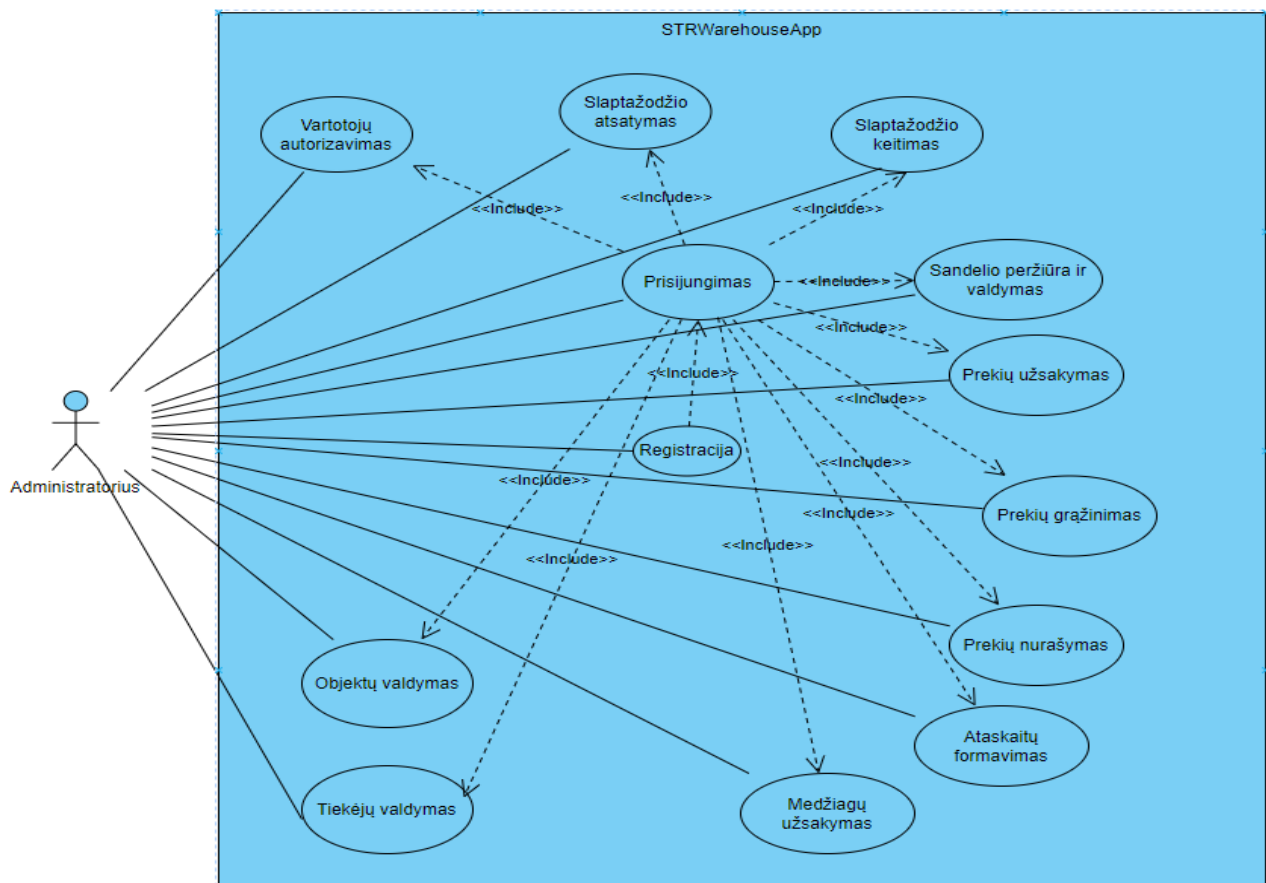
Programos specifikacija buvo ruošiamą atsižvelgiant į užsakovo poreikius sistemai. Nustatytos pagrindinės sistemos funkcijos, kurios buvo skaidomos į žemesnio lygio specifikacijos detalizacijas.

*Pagrindinės sistemos funkcijos:* Vartotojų valdymo funkcionalumas; Sandėlio peržiūra; Tiekėjai; Objektai; Prekių grąžinimas; Prekių užsakymas; Medžiagų užsakymas; Ataskaitų formavimas; Prekių nurašymas.

Duomenų saugai užtikrinti panaudotas SSL protokolas. Prisijungimui reikalingas prisijungimo vardas ir slaptažodis, kurie nėra viešai prieinami ir nematomi net sistemos vartotojui. Prisijungimo eilutę galima keisti tik iš aplikacijos konfigūracijos, tam reikalingas aplikacijos kodas ir iš naujo sukompiliuota programa. Svarbu paminėti, jog „GitHub“ laikmena nėra viešai prieinama, todėl prieiga prie serverio nebus matoma niekam, kas neturi teisės jos matyti ar konfigūruoti. Atsitikus atvejui, jei duomenų bazės lentelė būtų nutekinta, vartotojų slaptažodžiai yra koduoti SHA-256 algoritmu. Tai daroma tam, kad nutekinus duomenis slaptažodžio nebūtų įmanoma atkurti, kadangi dažnu atveju vartotojas gali naudoti tą patį slaptažodį ne vienoje sistemoje.

## Sistemos projektavimas panaudos atvejų diagrama

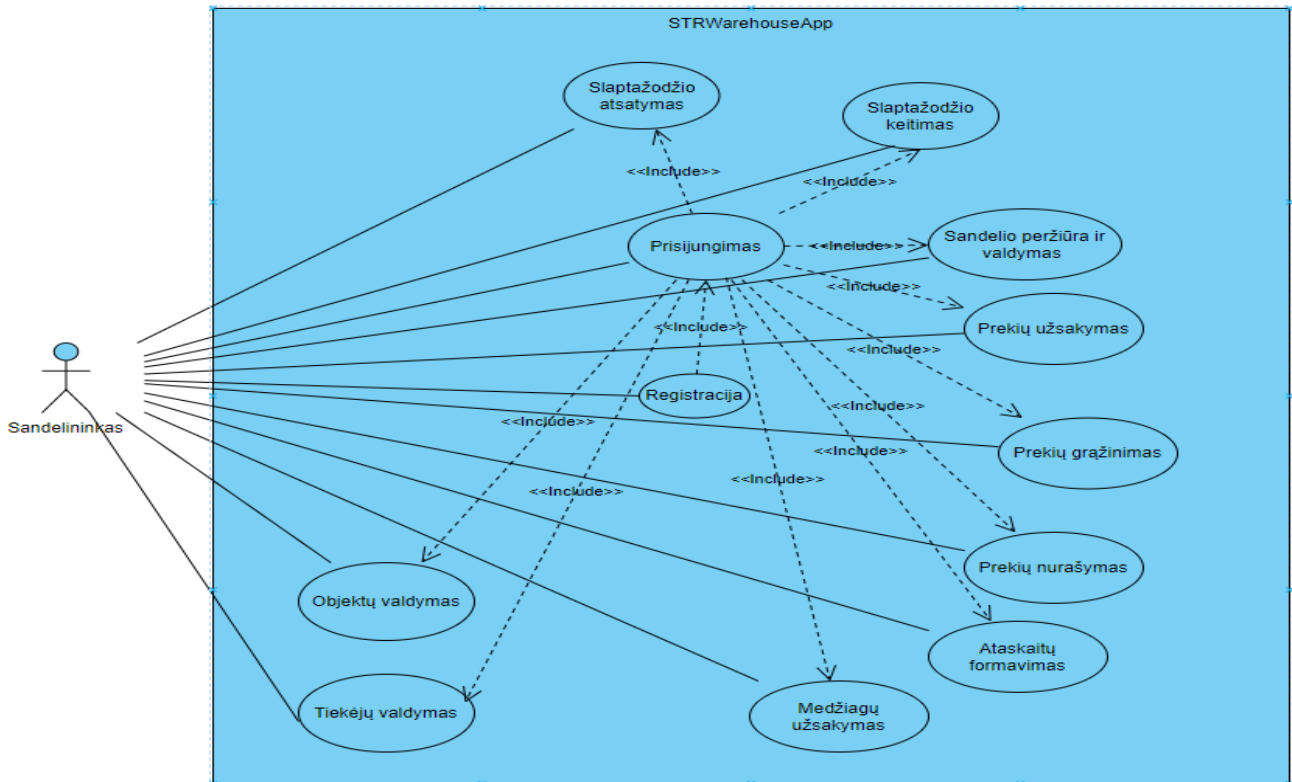
Panaudos atvejų diagramos iliustruoja ir apibūdina kontekstą bei reikalavimus visai sistemai arba sistemos daliai. Galima sumodeliuoti didelę ir komplikuoatą sistemą vienoje diagramoje arba daryti jų daugiau konkreitiems sistemos komponentams. Paprastai panaudos atvejų diagramos būna kuriamos ankstyvojoje sistemos vystymo fazėje ir toliau vystant sistemą yra remiamasi panaudos atvejų diagrama [15].



2 pav. Administratoriaus panaudos atvejai sistemoje



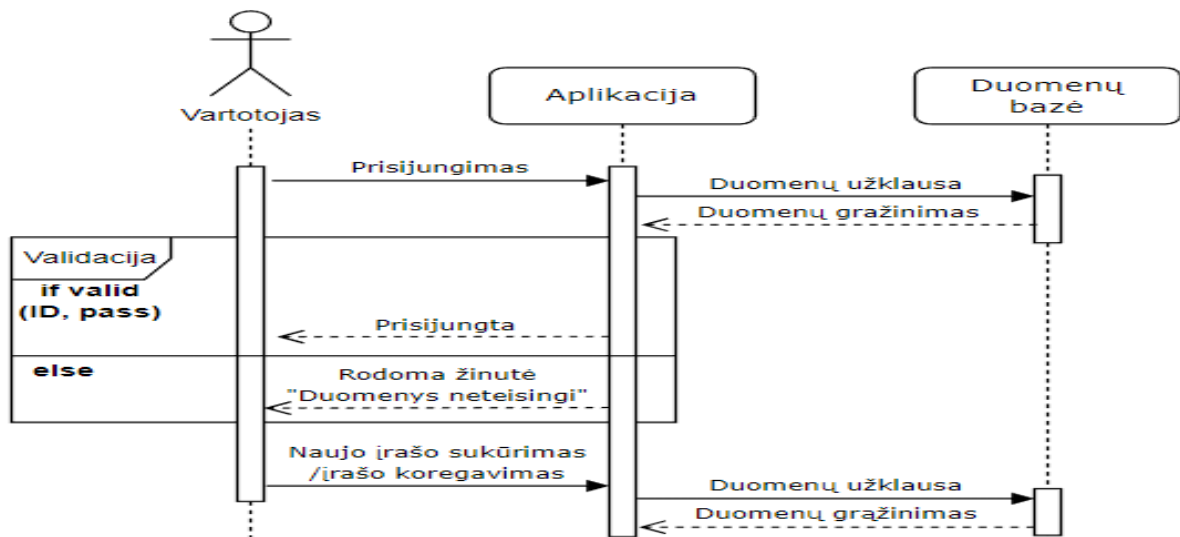
Kadangi sistemoje reikalingi 4 vartotojai, panaudos atvejai kuriami 3 iš jų, nes modeliavimo procese kliento nurodytoms vartotojų rolėms direktorius ir administratorius turės tas pačias galimybes sistemoje. Detalizuojami administratoriaus, sandėlininko ir darbų vadovo panaudos atvejai. Administratorius gali naudotis visomis sistemoje palaikomomis funkcijomis (3 pav.). Sandėlininkas yra pagrindinis sistemos vartotojas, kuris prižiūri sandėlį, patvirtina išdavimus ir gražinimus pagal faktą, kuruoja tiekėjų ir objektų sąrašus. Darbų vadovas yra vartotojas, kuris turi savo asmeninį prekių sandėlį, į kurį prekės gali patekti iš pagrindinio sandėlio, jį gali vykdyti užsakymus, gražinimus ir nurašymus, tačiau negali redaguoti pagrindinio sandėlio įrašų.



3 Pav. Sandėlininko panaudos atvejai sistemoje

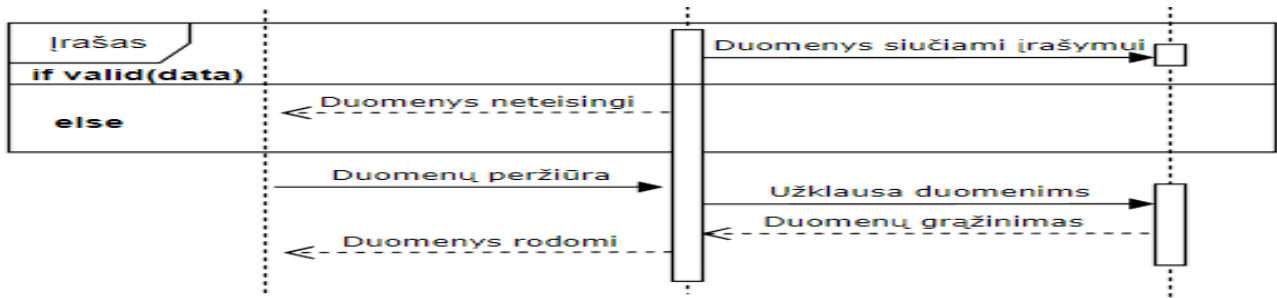
### Sekų diagramos detalizavimas

Kaip teigiama oficialioje IBM svetainėje [16], sekų diagrama yra viena iš UML diagramų, iliustruojanti sistemos seką tarp objektų. Sekų diagramą sudaro objektų grupė, objektai turi savo gyvavimo linijas ir žinutes, kuriomis keičiamasi tarp objektų dirbant su sistema.



4 pav. Sekų diagrama, 1dalis

Sekų diagramoje pavaizduota sąveika tarp objektų: Vartotojas; Aplikacija; Duomenų bazė.

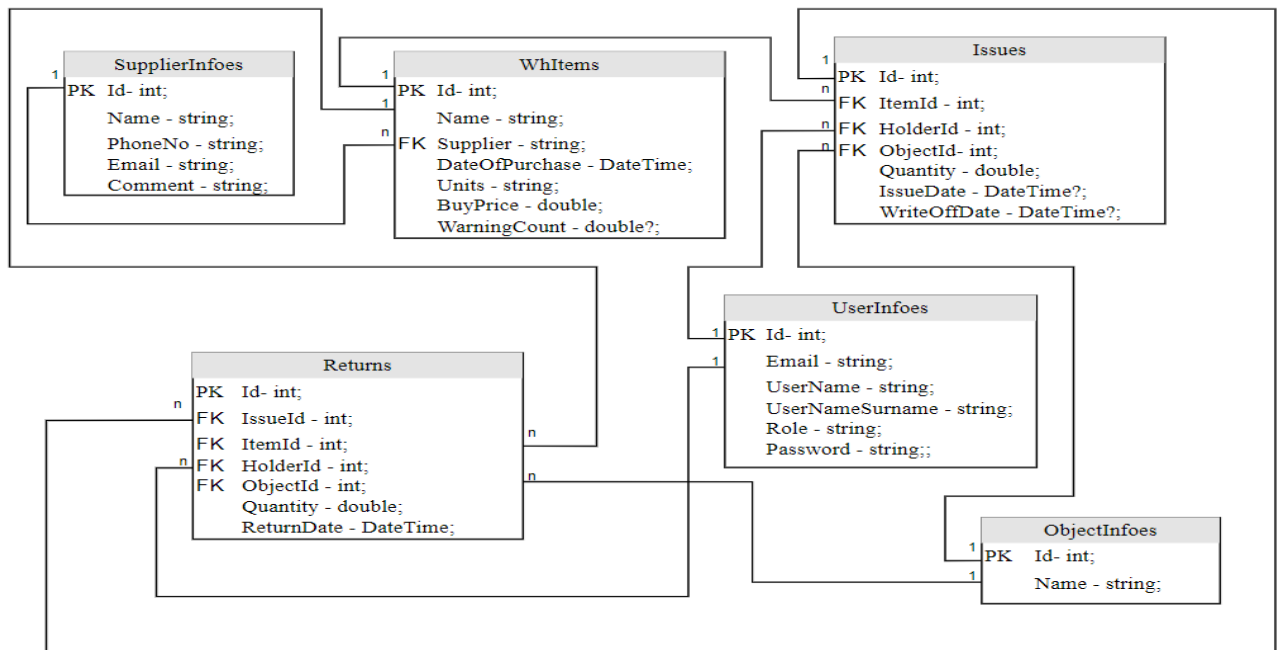


5 pav. Sekų diagrama, 2 dalis

Šiose diagramose matome standartinę objektų sąveiką, kurioje visi duomenų tikrinimo metodai vyksta aplikacijoje. Kiekvienas įrašo įrašymas ar koregavimas vyksta tokiu pat principu, išskyrus prisijungimo prie sistemos funkcionalumą, šiuo atveju tikrinimas vyksta sistemoje pagal įvestus prisijungimo duomenis. Jei duomenys neteisingi ar negalimi, aplikacija nesijungia su duomenų baze, sąveika tarp aplikacijos ir duomenų bazės įrašant ar modifikuojant įrašus veikia tik praėjus vidiniams aplikacijos patikrinimams. Duomenų rodymo metu užklausa siunčiamos prieš atvaizduojant duomenis aplikacijos formų lentelėse.

### Duomenų bazės projektavimas

Duomenų bazė yra struktūrizuotos informacijos duomenų rinkinys, laikomas kompiuterio saugykloje (serveryje). Duomenų bazę įprastai valdo duomenų bazės valdymo sistema (DBVS) [10].



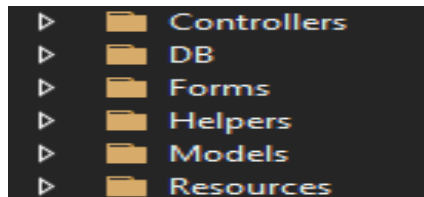
6 pav. Duomenų bazės schema

Duomenų bazė modeliuota pagal kliento poreikius sistemai ir saugomai informacijai laikyti bei kiek įmanoma mažiau dubliuoti duomenis. Pagrindinė duomenų bazės lentelė yra „WhItems“, kurioje saugomi faktiniai pagrindinio sandėlio likučiai, „Issues“ lentelėje saugomi prekės išdavimo vartotojui įrašai. „Returns“ lentelėje saugomi grąžinimai iki daiktų grąžinimo į sandėlį patvirtinimo, tai yra tarpinė lentelė iki prekių priėmimo patvirtinimo fakto. Kai įrašai patvirtinami, lentelės įrašas ištrinamas, o grąžintos prekės kiekis sumuojamas prie „WhItems“ lentelės atitinkamo įrašo pagal „Returns“ lentelės „ItemId“ elementą. Analogiškai ištrinamas Issues lentelės įrašas arba minusuojamas įrašo kiekis, jei grąžinama ne viską. „UserInfos“ lentelėje saugomi vartotojų duomenys. Slaptažodžio keitimo galimybė sistemoje palikta tik vartotojui, administratorius negali keisti slaptažodžio savo nuožiūra, visi slaptažodžiai koduoti SHA256 algoritmu ir nėra matomi atviru

tekstu. „SupplierInfos“ lentelėje saugomi duomenys apie tiekėjus, toliau tiekėjai naudojami tik „WhItems“ lentelėje įrašant „Supplier“ elementą. Tai buvo įgyvendinta tam, kad nesikeistų tiekėjų pavadinimai ir vartotojas sistemoje galėtų naudotis paieška ir filtravimu pagal tiekėją kuo efektyviau. „ObjectInfos“ lentelėje saugomi objekto duomenys, tai yra vidiniai įmonės objektų pavadinimai, veikimo principas sistemoje toks pat, kaip ir tiekėjų lentelės. Schemoje nurodyti lentelių pavadinimai yra techninės reikšmės, matomi lentelių pavadinimai vartotojui aplikacijoje yra kitokie, kai kur duomenys nėra saugomi, o apskaičiuojami aplikacijoje.

## Aplikacijos realizavimas

Aplikacija realizuota naudojant objektinio programavimo principus. Visi failai suskirstyti į paketus pagal kategoriją. Paketų kategorijos pateiktos 8 paveiksle.



7 pav. Aplikacijos paketų struktūra

Kiekvienas paketas laiko tam tikrus failus su tam tikromis klasėmis. Tai yra daroma tam, kad lengviau būtų atsekti kodą pagal grupę ar jo funkcionalumą bei paskirtį. Toliau detalizuojami paketai:

*Controllers* – naudojamas pagrindinėms sistemos kontroliavimo klasėms sudėti. Kontroliavimo klasės atsakingos už esminę sistemos veiklą pagrindiniame lygmenyje (core). Kontroliavimo klasėje aprašomi “Entity” karkaso metodai, kurie yra pritaikyti konkrečiam sistemos veiksmui atlikti ir susideda iš daugiau nei vieno numatytojo metodo, skirti sąveikai su duomenų baze. Inicijuojamos duomenų bazės lentelės, statiški aplikacijos kintamieji, kurie sekami per kontroliuojančios klasės instanciją.

*DB* – pakete sudėti failai, kuriuose yra klasės, susijusios su duomenų baze. Numatytieji “Entity” karkaso metodai, duomenų bazės kontekstas, kuris skirtas prisijungti ir pažinti konkrečią duomenų bazę, jei duomenų bazė pakeičiama ir bandoma pasiekti jau sukurtą duomenų bazę, jei reikalingi migracijos metodai, kurie aprašomi ir talpinami šiame pakete.

*Forms* – paketas skirtas „WinForms“ formų klasėms, dizainams ir kiekvienos formos veikimo logikai aprašyti. Juose nurodoma, kaip atrodys kiekviena forma, ką darys kiekvienas interaktyvus objektas aplikacijoje.

*Helpers* – pakete sudėtos klasės, kurios yra pagalbinės ir gali būti viešai naudojamas visur, kur ir kada jų reikia. Pagalbinės klasės ir metodai yra tokie: SMTP funkcionalumas, teksto kodavimo funkcionalumas, slaptažodžių generavimo funkcionalumas ir kt.

*Models* – paketas naudojamas laikyti aplikacijai reikalingiems modeliams, ten laikomi visų sistemoje naudojamų objektų modeliai, taip pat ir DB lentelių modeliai.

*Resources* – tai numatytasis .NET karkaso paketas, kuriame laikomi paveikslėliai, fiksuoti tekstai, kurie gali būti panaudojami globaliam sistemos vertimui į kitą kalbą, piktogramos, spalvų kodai ir kt.

Aplikacijos programavimo procesui palengvinti buvo naudojamos tam tikros bibliotekos ir paketai, kurie yra teikiami per oficialią Microsoft paketų diegimo sistemą NuGet [11]. Šie paketai yra perduodami kartu su aplikacijos instaliacija, kadangi be jų bibliotekų failų sistema gali neveikti, arba veikti nenuspėjamai. Visada lieka galimybė, kad vartotojas jau turi tam tikras bibliotekas įsidieges kartu su kitomis aplikacijomis, tačiau negalima to žinoti ir dar sudėtingiau žinoti lokacijas, kuriose yra šios bibliotekos, jei jos nėra sisteminės. Tam, kad išvengti aplikacijos veikimo sutrikimų, kiekviena panaudota biblioteka įrašoma kartu su aplikacija į jos aplanką. Senesnėse operacinėse sistemose jos turi senesnius .NET karkasus, kurie gali paveikti sistemos sutrikimus. Naudojami perskirstomieji karkasai, tačiau šiuo atveju aplikacija skirta „Windows“ 10 operacinei sistemai, kuri jau turi .NET 4.7.2 karkasą, naudojamą aplikacijai programuoti. Aplikacijos programavimui panaudoti paketai pateikti 9 paveiksle.

```
<packages>
<package id="BouncyCastle" version="1.8.5" targetFramework="net472" />
<package id="EntityFramework" version="6.4.4" targetFramework="net472" />
<package id="Google.Protobuf" version="3.14.0" targetFramework="net472" />
<package id="K4os.Compression.LZ4" version="1.2.6" targetFramework="net472" />
<package id="K4os.Compression.LZ4.Streams" version="1.2.6" targetFramework="net472" />
<package id="K4os.Hash.xxHash" version="1.0.6" targetFramework="net472" />
<package id="Microsoft.Bcl.AsyncInterfaces" version="1.1.0" targetFramework="net472" />
<package id="Microsoft.Bcl.HashCode" version="1.1.0" targetFramework="net472" />
<package id="Microsoft.EntityFrameworkCore" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.EntityFrameworkCore.Analyzers" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.EntityFrameworkCore.Relational" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Caching.Abstractions" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Caching.Memory" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Configuration" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Configuration.Abstractions" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Configuration.Binder" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.DependencyInjection" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.DependencyInjection.Abstractions" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Logging" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Logging.Abstractions" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Options" version="3.1.1" targetFramework="net472" />
<package id="Microsoft.Extensions.Primitives" version="3.1.1" targetFramework="net472" />
<package id="MySQL.Data" version="8.0.28" targetFramework="net472" />
<package id="MySQL.Data.EntityFramework" version="8.0.28" targetFramework="net472" />
<package id="System.Buffers" version="4.5.1" targetFramework="net472" />
<package id="System.Collections.Immutable" version="1.7.0" targetFramework="net472" />
<package id="System.ComponentModel.Annotations" version="4.7.0" targetFramework="net472" />
<package id="System.Diagnostics.DiagnosticSource" version="4.7.0" targetFramework="net472" />
<package id="System.Memory" version="4.5.4" targetFramework="net472" />
<package id="System.Numerics.Vectors" version="4.5.0" targetFramework="net472" />
<package id="System.Runtime.CompilerServices.Unsafe" version="5.0.0" targetFramework="net472" />
<package id="System.Threading.Tasks.Extensions" version="4.5.2" targetFramework="net472" />
<package id="SSH.NET" version="2016.1.0" targetFramework="net472" />
</packages>
```

8 pav. „NuGet” ir sisteminiai paketai, kurie yra panaudoti sistemos kūrimo metu

Realizuojant aplikacijos funkcionalumą buvo susidurta su tam tikromis problemomis, kurios buvo sprendžiamos atsižvelgiant į situaciją ir pasitelkiant nestandartinius sprendimo būdus. Nors šie būdai tikrai nėra vienintelis variantas išspręsti problemą, tačiau jie veikia ir atlieka savo funkcijas. Toliau pateikiami problemų pavyzdžiai ir jų sprendimo būdai.

### Problema Nr. 1:

Sistemai reikalingas el. laiško siuntimas, kai vartotojas pasikeičia slaptažodį ar generuojamas naujas slaptažodis iš administratoriaus formos.

### Sprendimas:

```
2 references
public class Smtplib
{
    2 references
    public static void SendMail(string subject, string body, string recipient)
    {
        string fromAddress = "str.support@inbox.lt";
        string password = ;
        SmtplibClient smtpClient = new SmtplibClient("mail.inbox.lt", 587);
        smtpClient.UseDefaultCredentials = false;
        smtpClient.Credentials = new NetworkCredential(fromAddress, password);
        smtpClient.EnableSsl = true;
        MailMessage mailMessage = new MailMessage(fromAddress, recipient, subject, body);
        smtpClient.Send(mailMessage);
    }
}
```

9 pav. Problemos Nr. 1 sprendimo kodo detalizavimas, SMTP

Funkcionalumui įgyvendinti panaudotas mail.inbox.lt serveris, kuris palaiko galimybę siųsti el. laiškus su tam tikrais apribojimais ne daugiau 15 laiškų per valandą ir ne daugiau 5 laiškų per minutę. Tokie apribojimai tenkina sistemos poreikius.

### Problema Nr. 2:

Rikiuojant įrašus grafinės sąsajos lentelėse reikalingas atsiminimas esamai instancijai apie rikiavimo statusą: sistema turi žinoti, kada buvo išrikiuota didėjimo tvarka, kad galėtų kitu paspaudimu rikiuoti mažėjančia tvarka.

### Sprendimas:

```

1 reference
private void LVWarehouseCollumnClicked(object sender, ColumnClickEventArgs e)
{
    // Sort the items in the ListView control by the clicked column
    var column = e.Column;
    var items = LVWarehouse.Items.Cast<ListViewItem>().ToList();

    switch (column)
    {
        case 0:
            if (firstCollumnClicked)
            {
                //Ascending
                items.Sort((x, y) => int.Parse(x.SubItems[column].Text).CompareTo(int.Parse(y.SubItems[column].Text)));
                firstCollumnClicked = false;
            }
            else
            {
                //Decending
                items.Sort((x, y) => int.Parse(y.SubItems[column].Text).CompareTo(int.Parse(x.SubItems[column].Text)));
                firstCollumnClicked = true;
            }
            break;
        case 1:
            if (secondCollmunClicked)

```

10 pav. Problemos Nr. 2 sprendimo kodo detalizavimas, rikiavimo statusas

Rikiavimo statusas išsaugomas esamos instancijos „bool“ tipo kintamajame, taip panaudojant switch-case funkcionalumą pasiekiamas kiekvieno stulpelio rikiavimo funkcionalumas.

### Problema Nr. 3:

Tiekėjo pavadinimas ir objekto pavadinimas turi būti įvedamas atviru tekstu, tačiau turi visiškai atitikti įrašą lentelėje.

### Sprendimas:

```

suppliers = Controller.Instance.GetAllSuppliers();
foreach (SupplierInfo supplier in suppliers)
{
    tBSupplier.AutoCompleteCustomSource.Add(supplier.Name);
}
tBSupplier.TextChanged += tBSupplier_TextChanged;

```

11 pav. Problemos Nr. 3 sprendimo kodo detalizavimas, užpildomas automatinis užbaigimo sąrašas

12 paveiksle pavaizduota kodo iškarpa, kodas gauna visus registruotus tiekėjus ir užpildo jais automatinio užbaigimo sąrašą.

```

1 reference
private void tBSupplier_TextChanged(object sender, EventArgs e)
{
    string currentText = tBSupplier.Text;
    string matchingItem = tBSupplier.AutoCompleteCustomSource.Cast<string>()
        .FirstOrDefault(item => item.Equals(currentText, StringComparison.CurrentCultureIgnoreCase));

    if (matchingItem != null)
    {
        int selectionStart = tBSupplier.SelectionStart;
        int selectionLength = tBSupplier.SelectionLength;

        string newText = matchingItem;
        tBSupplier.Text = newText;
        tBSupplier.SelectionStart = selectionStart;
        tBSupplier.SelectionLength = selectionLength;
    }
}

```

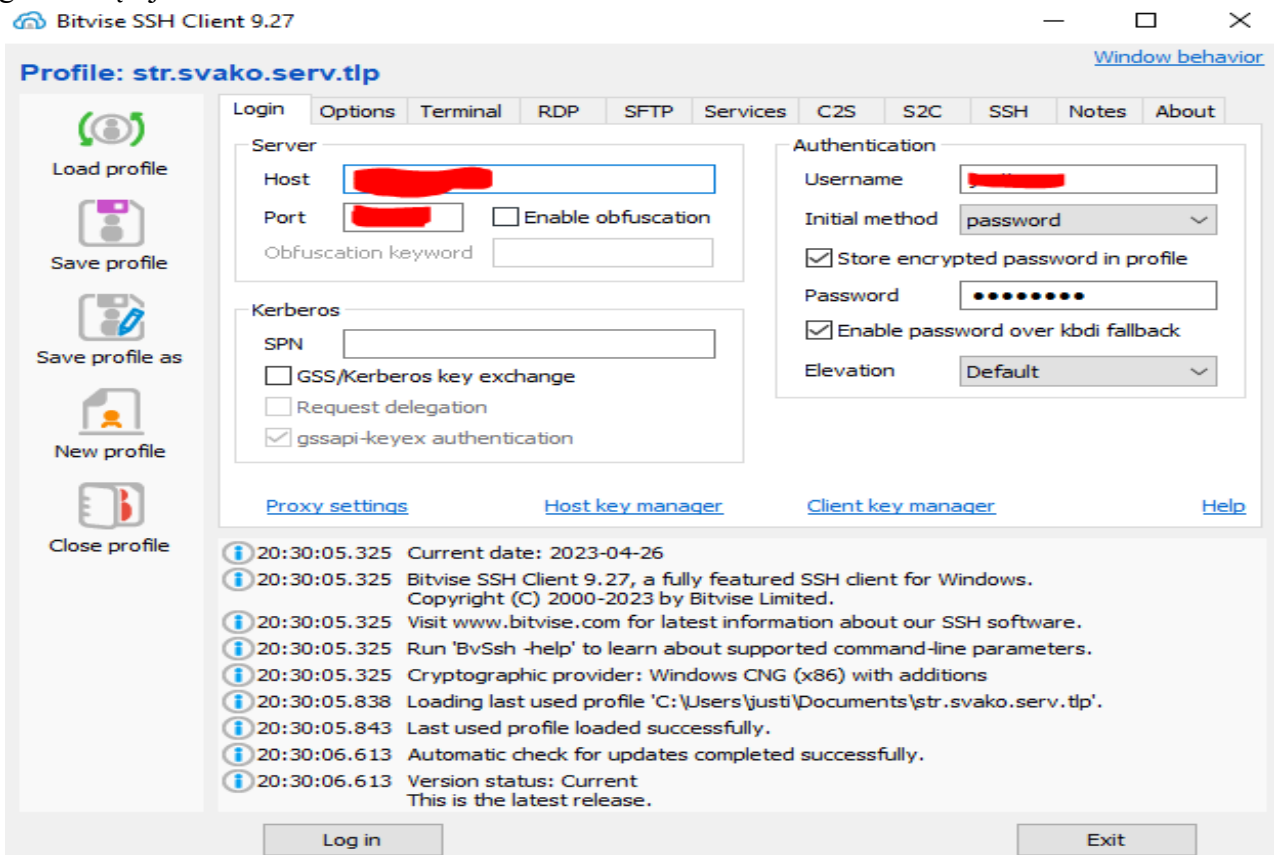
12 pav. Problemos Nr. 3 sprendimo kodo iškarpa, teksto pasiūlymas

13 paveiksle pavaizduotas metodas, kuris vartotojui siūlo pasirinkimą pagal įvestą tekstą, pvz., jei vartotojas pradeda rašyti „tiekėj...“, šis metodas automatiškai randa visus tiekėjus, kurie yra registruoti ir atitinka pradėtą įvesti tekstą, pvz. „Tiekėjas 1“, „Tiekėjas 2“, „Tiekėjas 3“. Taip automatiškai pasiūlomas tekstas, užbaigti įrašą galima tiesiog paspaudus **TAB** mygtuką, tokiu atveju teksto laukelis bus užpildytas pirmuoju pasiūlymu. Kitą pasiūlymą galima rinktis rodyklių arba kompiuterio pelės pagalba. Toliau vyksta tikrinimas, ar įvestas tiekėjo pavadinimas atitinka

registruoto tiekėjo pavadinimą. Jei pavadinimas neatitinka, parodomas pranešimas su tekstu „Toks tiekėjas nerastas” ir sistema neleidžia pridėti įrašo. Taip išvengiama neatitikimų su tiekėjo pavadinimo įvedimu. Objekto įvedimas veikia tokiu pat principu.

## Duomenų bazės realizavimas

Duomenų bazė realizuota SVAKO vidiniame virtualiame serveryje. Prisijungimas prie serverio vykdomas per „Bitvise SSH Client” (žr. 14 pav.) aplikaciją, kuri turi daug įrankių vienoje vietoje. Prie serverio ir duomenų bazės galima prisijungti per komandinę eilutę arba terminalą, ši programa naudojama dėl paprastesnio funkcionalumo, išsaugomų prisijungimo duomenų, failų valdymo su grafine sąsaja.



13 pav. „Bitvise SSH Client” grafinė sąsaja

Suprogramuotoje aplikacijoje duomenų bazės modelių ir lentelių iniciavimas vykdomas panaudojant „Entity” karkasą.

```
1 reference
public Controller()
{
    dbContext = new WhDbContext();
    whItemRepository = new GenericRepository<WhItem>(dbContext);
    userRepository = new GenericRepository<UserInfo>(dbContext);
    supplierRepository = new GenericRepository<SupplierInfo>(dbContext);
    issuesRepository = new GenericRepository<Issue>(dbContext);
    returnsRepository = new GenericRepository<Return>(dbContext);
    objectsRepository = new GenericRepository<ObjectInfo>(dbContext);
}
```

14 Pav. „Controller” klasės konstruktorius

Iniciacija vykdoma „Controller” klasės konstruktoriuje (žr. 15 pav.). Toliau visos lentelės pasiekiamos per nurodytus kintamuosius. Kiekviena lentelė turi savo numatytuosius metodus, kuriuos paveldi iš klasės „GenericRepository”: Insert; Update; Delete; Save; GetById.

Numatytieji metodai ne visais atvejais yra pakankami sudėtingesnėms operacijoms atlikti, todėl buvo kuriami specifiniai metodai, kurie kiekvienu atveju buvo programuojami pagal poreikį.

```

1 reference
public bool ReturnItem(Return returnItem)
{
    List<Return> returns = returnsRepository.Get().ToList();
    Issue issue = issuesRepository.GetByID(returnItem.IssueId);

    Return theReturn = null;
    foreach (Return ret in returns)
    {
        if (ret.HolderId == returnItem.HolderId && ret.ItemId == returnItem.ItemId)
        {
            theReturn = ret;
            break;
        }
    }

    double quantity = 0;

    if (theReturn != null)
        quantity = theReturn.Quantity;

    if (issue.Quantity < (quantity + returnItem.Quantity))
        return false;

    if (theReturn != null)
    {
        theReturn.Quantity += returnItem.Quantity;
        theReturn.ReturnDate = DateTime.Now;
        returnsRepository.Update(theReturn);
    }
    else
    {
        returnsRepository.Insert(returnItem);
    }
    returnsRepository.context.SaveChanges();
    return true;
}

```

15 pav. „ReturnWhItem” metodo kodo iškarpa

Pateiktas pavyzdys (žr. 16 pav.), kuriame atliekama prekės grąžinimo operacija. Šiame metode visų pirma patikrinama, ar grąžinamos prekės įrašas jau egzistuoja. Jei jis randamas, sistema atnaujina įrašą, pridėdama grąžinamos prekės kiekį prie jau esamo įrašo ir atnaujina grąžinimo datą. Kitu atveju kuriamas visiškai naujas įrašas.

## Sistemos testavimas

Kuriant aplikaciją ir sąveiką su kitomis sistemos dalimis, duomenų bazės testavimas buvo vykdomas viso kūrimo proceso metu. Galutinis sistemos testavimas buvo atliekamas pasitelkiant metodiką, aprašomą Thomas Hamilton straipsnyje [6]. Sistemos funkcionalumas testuojamas pagal straipsnyje nurodomą formatą aprašant testavimo atvejus, testavimo duomenis, sužadavimo veiksmus ir tikėtiną rezultatą. Atlikti 46 testavimo atvejai, pavyzdys pateiktas 8 lentelėje.

### 7 lentelė. Testavimo atvejo “TC-01” registracijos funkcionalumui testuoti pavyzdys

Testavimo atvejo Id	Aprašymas	Sužadavimo veiksmai	Testavimo duomenys	Tikimasis rezultatas	Gautas rezultatas	Pavyko / Nepavyko
TC-01	Registracija į sistemą	Įjungiamą aplikaciją, paspaudžiamas mygtukas registruotis, įvedami registracijos duomenys, paspaudžiamas mygtukas “Registruotis”.	Prisijungimo vardas = test, Slaptažodis = test, El. Pašto adresas = Justinas.zatorskis@stud.svako.lt	Vartotojas gauna savo prisijungimo instrukcijas į nurodytą el. Pašto adresą, vartotojas prisiregistravęs sistemoje.	Vartotojas gauna savo prisijungimo instrukcijas į nurodytą el. Pašto adresą, vartotojas prisiregistravęs sistemoje.	Pavyko

Testavimo planas buvo skirstomas į atskirus testavimo siekius, rezultatus ir orientuotas į skirtingas sistemos aplikacijos dalis. Testuotas aplikacijos aptikimas su skirtingomis antivirusinėmis programomis, pasitelkus „VirusTotal” testavimo platformą [19]. Automatizuotas aplikacijos

tikrinimas vykdomas pasitelkiant „Zenbox” virtualios mašinos testavimo įrankį, kuris suteikiamas „VirusTotal” platformoje nemokamai ir skirtas aptikti nepageidaujamus sistemos veiksmus. Kiekviena dalis, kuri buvo testuojama rankiniu būdu, buvo nagrinėjama pagal testavimo atvejus.

Aplikacijos testavimas suskirstytas į šias kategorijas:

- Autentifikacijos testavimas;
- Saugumo testavimas;
- Vartotojų valdymo testavimas;
- Pagrindinio sandėlio funkcionalumo testavimas;
- Medžiagų užsakymo testavimas;
- Ataskaitų formavimo testavimas;
- Objektų ir tiekėjų valdymo testavimas.

Testavimo metu buvo tikrinamos funkcijos, kurios dirba su duomenimis, pagalbinės funkcijos, ieškoma kodo klaidų. Rastos klaidos buvo iš karto taisomos „Agile“ principu.

Atlikus testavimą, buvo pastebėta, jog sistemos kaip vieneto funkcionalumas atitinka reikalavimus, duomenys redaguojami, įrašomi ir trinami pagal vartotojo teises sistemoje, neatitikimų nerasta, sistema „nelūžta“ naudojant aplikaciją. Testavimo metu buvo pastebėta, jog sistema neveikia vietiniame tinkle kartu su serveriu. Tai įvyksta dėl to, kad, jungiantis iš vietinio tinklo prie serverio, reikia jungtis naudojant vietinį IP adresą, kuris nustatomas tinklo konfigūracijose. Tačiau šis atvejis nėra aktualus dėl to, kad sistema nėra skirta jungtis per LAN. Vartotojai iš išorinio tinklo prisijungia be nesklandumų.

## **Išvados**

Užsakovo poreikių ir įmonės analizei pasitelktas interviu metodas, interviu metu buvo išsiaiškinti reikalavimai sistemai ir įmonėje naudojami informacinių technologijų įrankiai. Į kintamus poreikius buvo atsižvelgiama „Agile“ principu, atsirandant papildomiems poreikiams ir idėjoms, jos buvo įgyvendinamos sistemos kūrimo procese.

Atliktoje panašių sistemų analizėje buvo palygintos 3 panašios sistemos pagal D. Luther [1] siūlomus ERP sistemos vertinimo kriterijus.

Lyginant galimas kūrimo technologijas, buvo atsižvelgiama į reikalavimus kuriamai sistemai, sukurtas veikiantis prototipas įmonės naudojamoje operacinėje sistemoje. Palygintos kūrimo technologijos „JAVA“, „PHP“ ir „C#“. Pasirinkta „C#“ programavimo kalba. Sukurta „Winforms” aplikacija panaudojant „Entity framework” karkasą „MySQL“ duomenų bazei inicijuoti. Aplikacijai pagrindiniu karkasu pasirinktas „NET Framework 4.7.2“. Jis pasirinktas dėl to, kad užsakovo operacinė sistema „Windows 10 PRO“ turi šios versijos numatytąjį karkasą, todėl nereikalingi papildomai įrašomi persikirstomieji. Suprojektuotos UML diagramos, iliustruojančios sistemos veikimo principus, vartotojų panaudos atvejus ir duomenų bazės struktūrą. Nustatyti funkcionalumai sistemos naudojimui: administratorius turi visas galimybes naudotis visomis sistemos funkcijomis be išimčių. Pagrindinis papildomas funkcionalumas administratoriui yra tai, jog jis gali valdyti vartotojų paskyras, suteikti roles arba panaikinti galimybę vartotojui prisijungti prie sistemos. Visos likusios aplikacijos funkcijos pasiekiamos sandėlininko rolę turinčiam vartotojui. Sandėlininkas yra pagrindinis sistemos vartotojas, kuris kuruoja pagrindinį sandėlį, išdavimus, priėmimus, nurašymus ir pan.

Suprogramuota aplikacija, sistemos prototipas „STRWarehouseApp“, kuris duomenis saugo ir apdoroja „MySQL“ duomenų bazėje. Įdiegtas funkcionalumas automatiniam el. laiškų siuntimui iš aplikacijos, panaudotas „SMTP“ klientas iš serverio „inbox.lt“. Sistemos serveris yra sukurtas SVAKO tinkle kaip virtualus serveris, jame įdiegta „MySQL“ duomenų bazė, kuri inicijuota ir aprašyta aplikacijos kode, pasitelkus „Entity framework“ karkasą.

Sistemos prototipo testavimas buvo atliekamas viso programavimo proceso metu, galutinis testavimas atliktas pasitelkus testavimo atvejus ir metodiką, aprašytą T. Hamilton [6]. Visos rastos klaidos buvo taisomos iškart jas pastebėjus ir testuojamos pagal faktą.



## Literatūra

1. Luther, D. *10 Criteria to Select and Compare ERP System Vendors*. Oracle NetSuite. (2022). <https://www.netsuite.com/portal/resource/articles/erp/erp-vendor-selection-criteria.shtml>
2. *What is ERP?* Oracle. (2023). <https://www.oracle.com/erp/what-is-erp/>
3. *Entity Framework overview*. Microsoft. (2021). <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>
4. *Overview of .NET Framework*. Microsoft. (2023). <https://learn.microsoft.com/en-us/dotnet/framework/get-started/overview>
5. *What is NetSuite ERP?* Oracle NetSuite. (2023). <https://www.netsuite.com/portal/products/erp.shtml>
6. Hamilton, Th. *What is a Test Case?* Guru99. (2023). <https://www.guru99.com/test-case.html>
7. Ali, M., and Miller, L. ERP system implementation in large enterprises – a systematic literature review. *Journal of enterprise information management*, 30(4). (2017): 666–692.
8. Šiaulių tauro ryšiai. (2023). <https://stryšiai.lt/>
9. *A Tour of C# language*. Microsoft. (2023). <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
10. Daugirdas, D., Pranciulis, L., and Čepis, A. Projection and implementation of a model for a game rental system. *Applied Scientific Research* 2(1). (2023): 65–73.
11. Beaumont, A., and Bastaki, B. B. An Investigation into the Prevalence of Reflection Techniques in Distributed Microsoft. Net NuGet Artefacts. In *ICSCA 2022: 11th International Conference on Software and Computer Applications*, pp. 173-178. (2022). <https://doi.org/10.1145/3524304.3524329>
12. *Advantages and Disadvantages of PHP*. GeeksforGeeks. (2023). <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-php/>
13. *Advantages and Disadvantages of Java*. TechVidvan. (2023). <https://techvidvan.com/tutorials/pros-and-cons-of-java>
14. Rini, N. *C# vs Java*. Codeguru. (2023). <https://www.codeguru.com/csharp/c-sharp-vs-java/>
15. *Use-Case Diagrams*. IBM. (2021). <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>
16. *Sequence Diagrams*. IBM. (2021). <https://www.ibm.com/docs/sk/rsm/7.5.0?topic=uml-sequence-diagrams>
17. *Overview of Dynamics 365*. Microsoft. (2023). <https://learn.microsoft.com/en-us/dynamics365/get-started/intro-crossapp-index>
18. *Specialized ERP software for real estate & construction industry*. StrategicERP. (2023). <https://strategicerp.com/>
19. VirusTotal. (2023). <https://www.virustotal.com/gui/home/upload>