

**Kauno technologijos universitetas**

Informatikos fakultetas

## **Atostogų apskaitos informacinės sistemos kūrimas**

Baigiamosios praktikos ataskaita

---

**Giedrė Pranauskytė**

Studentė

**Gintautas Kučinskas**

Vadovas

**Doc. Giedrius Ziberkas**

Dėstytojas

---

## Turinys

<b>Ivadas.....</b>	<b>3</b>
<b>1. Analizės dalis.....</b>	<b>4</b>
1.1. Informacija apie įmonę.....	4
1.1.1. Įmonės veiklos sritis, specializacija.....	4
1.1.2. Įmonės vizija.....	4
1.1.3. Įmonės misija.....	4
1.1.4. Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją.....	4
1.2. Informacija apie įmonėje naudojamą IT.....	4
1.3. Įmonės veiklos procesai.....	4
1.3.1. Projekto inicijavimas.....	4
1.3.2. Projekto koncepcijos analizė.....	5
1.3.3. Kainos ir investicijų analizė.....	5
1.3.4. Projektų vykdymas, pristatymas ir dokumentacija.....	5
1.3.5. Komandinis darbas.....	5
1.3.6. Inovacijos.....	5
1.3.7. Produktų vystymo procesai.....	5
<b>2. Praktikos dalis.....</b>	<b>6</b>
2.1. Techninė užduotis.....	6
2.1.1. Sistemos apibrėžimas.....	6
2.1.2. Bendras veiklos tikslas.....	6
2.1.3. Sistemos pagrįstumas.....	6
2.1.4. Konkurencija rinkoje.....	6
2.1.5. Prototipai ir pagalbinių informacija.....	7
2.1.6. Siūlomas sprendimas.....	7
2.1.7. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti.....	8
2.2. Reikalavimų specifikacija.....	9
2.2.1. Komercinė specifikacija.....	9
2.2.2. Sistemos funkcijos.....	9
2.2.3. Apribojimai.....	13
2.2.4. Duomenų ir rezultatų specifikacija.....	14
2.2.5. Vartotojo sąsajos specifikacija.....	15
2.2.6. Realizacijai keliami reikalavimai.....	17
2.2.7. Techninė specifikacija.....	18
2.3. Projektavimo eiga ir sistemos projektas.....	19
2.3.1. Projektavimo valdymas ir eiga.....	19
2.3.2. Projektavimo technologija.....	19
2.3.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos.....	19
2.3.4. Informacijos šaltiniai.....	20
2.3.5. Sistemos sudėtis.....	21
2.3.6. Komponentai.....	22
2.3.7. Sąsajos įvertinimas pagal vartotojo patyrimą, profesinę terminologiją.....	22
2.3.8. Duomenų kontrolė.....	24
2.4. Testavimo eiga ir rezultatai.....	25
2.5. Dokumentacija.....	30
<b>Išvados.....</b>	<b>31</b>
<b>Literatūra.....</b>	<b>32</b>

## Ivadas

Baigiamoji praktika buvo atliekama UAB „Xplicity“. Tai Olandijos įmonė, turinti komandas Amsterdamo ir Kaune. Įmonė užsiima kompiuterinių informacinių sistemų kūrimu ir pardavimu, naudoja įvairias IT technologijas, rengia mokymus ir akademijas, siekdami pasidalinti savo žiniomis ir patirtimi. Nors įmonė yra nedidelės struktūros, viskas atliekama labai organizuotai. Planavimas ir išsamus kiekvieno proceso dokumentavimas efektyvina įmonės darbą, todėl yra stipriai vertinami.

Atlikti praktiką šioje įmonėje buvo nuspręsta todėl, kad stengiamasi kuo daugiau bendrauti anglų kalba. Šitaip galima pagilinti minėtos kalbos gebėjimus profesinėje aplinkoje. Taip pat, kiekvienam praktikantui suteikiama proga ir netgi skatinama išbandyti kuo daugiau projektui naudojamų technologijų, kad kiekvienas baigęs praktiką vėliau turėtų įvairios patirties ir platesnes galimybes. Skatinamas komandinis darbas, bendradarbiavimas, tarpusavio pagalba, dirbama remiantis *Agile* metodologija. Vienintelė neigiama pusė yra ta, kad teko kurti naują produktą, o ne tobulinti seną, kas yra sudėtingiau ir būtų suteikę dar daugiau patirties, tačiau bent su dalimi panašių sunkumų buvo susidurta kai buvo pereinama prie naujesnių naudojamų karkasų versijų, perdaroma vartotojo sąsaja.

Praktikos tema – „Atostogų apskaitos informacinės sistemos kūrimas“. Bendras tikslas atliekant praktiką buvo susipažinti su įmonės veikla, patobulinti studijų programoje įgytas žinias, pritaikyti jas kuriant realų produktą ir susipažinti su reikalavimų bei kokybės valdymo ypatumais. Kalbant apie pačią sistemą, pagrindinis tikslas praktikos metu buvo minėti atostogų apskaitos informacinei sistemai realizuoti viršvalandžių posistemę, sukurti pirminius testus, aprašyti diegimo instrukcijas.

Prieš pradėdant praktiką buvo sudarytas planas, pagal kurį ir buvo atliekamas užduočių įgyvendinimas:

- Išanalizuoti įvairius į MS Visual Studio integruotus vienetų testų karkasus, ir, pasirinkus vieną, parašyti pirminius vienetų testus.
- Projekto repozitorijos pradiniam aplanke sukurti README.md failą, jame aprašyti projekto diegimo ir paleidimo instrukcijas.
- Sudaryti galimybę vartotojui panaudoti sukauptus viršvalandžius pateikiant naują atostogų prašymą. Administratorius gali fiksuoti sukaupytą darbuotojo viršvalandžių skaičių, šis paverčiamas į atostogų dienas, kurias darbuotojas gali panaudoti apmokamoms atostogoms.
- Pasirūpinti viršvalandžių panaudojimo validacija, teisingais skaičiavimais: patvirtinus atostogų prašymą, nuskaičiuojami panaudoti viršvalandžiai, kurie taip pat sutaupo įprastas atostogų dienas.
- Informaciją apie viršvalandžius įtraukti į el. laiškus, prašymo ir patvirtinimo dokumentus.
- Patobulinti vartotojo sąsają: įrašų lentelėje rikiavimas, skirtingas atitinkamo statuso dizainas.

Kas kartą baigus vieną iš užduočių, programinis kodas buvo keliamas į bendrą Azure repozitoriją ir peržiūrimas dėl galimų klaidų. Kad būtų sulietas į pagrindinę programą, turėdavo atitikti šiuos reikalavimus:

- neturi konfliktų su pagrindine programa,
- patvirtintas bent vieno iš praktikantų,
- patvirtintas bent vieno iš vyresniųjų programuotojų,
- sukompiliuojamas be klaidų,
- ištaisyti visi komandos narių pakomentuoti defektai,
- sėkmingi visi vienetų testai.

Toliau šioje ataskaitoje pateikiami įmonės analizė ir projekto, prie kurio buvo dirbta, analizė.

## **1. Analizės dalis**

### **1.1. Informacija apie įmonę**

#### **1.1.1. Įmonės veiklos sritis, specializacija**

Įmonės veiklos sritis – informacinės technologijos. „Xplicity“ užsiima kompiuterinių informacinių sistemų (žiniatinklio ir darbalaukio aplikacijų) kūrimu, priežiūra bei pardavimu. Paslaugos teikiamos pagal užsakymą, aplikacijos kuriamos įvairiomis programavimo kalbomis.

#### **1.1.2. Įmonės vizija**

Įmonės vizija – padėti savo klientams padidinti verslo vertę, įgyvendinti tikslus bei pasiekti to, kuo jie nori būti. „Xplicity“ tiki, kad jų sukurta programinė įranga priartina technologijas prie verslo ir padeda jų klientams įgyvendinti pažangius sprendimus.

#### **1.1.3. Įmonės misija**

Įmonės misija – teikti profesionalius programinės įrangos sprendimus savo klientams, tapti Lietuvos ir užsienio rinkose pripažinta įmone. „Xplicity“ stengiasi sukurti aplinką produktyviam darbui, išlaikyti skaidrumą, garantuoti savo darbuotojų atsakomybę ir kompetenciją.

#### **1.1.4. Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją**

Įmonė siekia turėti patyrusių ir nuolatos tobulėjančių specialistų komandą, galinčią susidoroti su sudėtingomis užduotimis ir savo klientų norus patenkinti kokybiškais darbais. „Xplicity“ kasmet vykdo įvairius mokymus, rengia dviejų savaitių akademiją, dalijasi patirtimi ir savo sukauptomis žiniomis su jaunais programuotojais, siekdami atrasti gabių, atsakingų ir motyvuotų potencialių darbuotojų. Geriausiai akademijoje pasirodžiusiems studentams suteikiami praktikos ar darbo pasiūlymai.

Įmonė turi keletą klientų ir orientuojasi į šias sritis: virtualaus darbalaukio infrastruktūra, telekomunikacijos, agrokultūra ir logistika. Puiki reputacija ir teigiami klientų atsiliepimai įmonei labai svarbūs – tai naujų klientų traukos šaltinis.

### **1.2. Informacija apie įmonėje naudojamas IT**

- Programavimo kalbos – C#, Java, JavaScript, Typescript.
- Mikroservisų programavimo karkasai – .NET, .NET Core.
- Vartotojų sąsajos programavimo karkasai – Angular, React.
- Mobilųjų programėlių programavimo karkasai – Xamarin.
- Integruotos programavimo aplinkos – Visual Studio Enterprise, JetBrains Ultimate.
- Operacinės sistemos – Windows, Linux.
- Versijavimo kontrolės įrankiai – Git, TFS.
- Agile proceso įrankiai – Pivotal tracker, Trello board, Microsoft Team Services.

### **1.3. Įmonės veiklos procesai**

#### **1.3.1. Projekto inicijavimas**

Klientui išsakius savo pageidavimus, įmonė juos įvertina ir pasiūlo visus galimus įgyvendinimo variantus. Pasirinkęs vieną iš jų, klientas pasirenka ir labiausiai darbui tinkamus programuotojus.

### **1.3.2. Projekto koncepcijos analizė**

Norėdama išsiaiškinti kliento poreikius, įmonė skiria pokalbį, kurio metu su klientu aptaria norimas matyti sistemos funkcijas, kūrimo eigą, galutinį pristatymo laiką ir priežiūros trukmę.

### **1.3.3. Kainos ir investicijų analizė**

Kiekvieno įmonės programuotojo valandiniai arba mėnesiniai įkainiai yra nustatyti pagal jo kompetencijos lygį. Klientas pats analizuoja kainas ir sprendžia, ar verta investuoti į paslaugą.

### **1.3.4. Projektų vykdymas, pristatymas ir dokumentacija**

Dauguma projektų vykdomi remiantis Agile metodologija, leidžiančia susiplanuoti darbų eigą, iteracijų trukmę ir aptarimą, komunikavimo būdus. Komandą įprastai sudaro projekto vadovas (dažniausiai kliento įmonės atstovas), penki programuotojai ir vienas testuotojas.

Projekto metu kuriama techninė dokumentacija, sudaryta iš galinio vartotojo vadovo, programuotojo darbo aplinkos paruošimo instrukcijų, API (aplikacijos programavimo sąsajos) dokumentacijos. Taip pat papildoma vaizdo įrašų biblioteka, kurioje saugomos programų naudojimo instrukcijos, informacija apie skirtingų modulių veikimą.

### **1.3.5. Komandinis darbas**

Komandinis darbas vyksta taikant Scrum (Agile metodą). Kiekviena komanda turi vadovą, kuris padeda komandos nariams ir rūpinasi, kad užduotys būtų atliktos laiku. Visos užduotys pagal prioritetą dokumentuojamos Backlog'e. Iteracijos (sprintai), kurių metu užbaigiamos kelios stambios užduotys, trunka dvi savaites. Sprinto pabaigoje rodoma užbaigtų užduočių demonstracija, išrenkamos tolesnės užduotys, bei vykdomas progreso aptarimas (retrospektyva). Retrospektyvos esmė – išanalizuoti praėjusio sprinto patirtį, suprasti, kodėl nepavyko įvykdyti suplanuotų užduočių.

### **1.3.6. Inovacijos**

Įmonė projektų kūrimui naudoja naujausias programavimo technologijas, nes tai darbą daro lengvesniu, našesniu ir patrauklesniu klientui. „Xplicity“ padeda įmonėms skaitmenizuoti savo procesus, palengvinti duomenų valdymą, leidžia įvertinti ir prognozuoti veiklos rezultatus. Pavyzdžiui, grūdų auginimu ir prekyba užsiimančioms įmonėms „Xplicity“ sukurta programinė įranga padeda sekti visą procesą nuo grūdų užauginimo iki pardavimo.

### **1.3.7. Produktų vystymo procesai**

Prieš pradėdant darbą, komanda aprašo kiekvienos funkcionalumo savybės reikalavimus į dokumentą, kurį vėliau naudoja tą konkrečią savybę realizuodami. Visas darbas dalijamas dviejų savaitinių iteracijomis, prieš naujos iteracijos pradžią sudaromas darbų sąrašas. Kiekvieną rytą visa komanda susitinka, parodo kas ką spėjo padaryti per praėjusią dieną, o dviejų savaitinių pabaigoje aptariamas bendras progresas, vdoma retrospektyva, sudaromas naujas užduočių sąrašas. Kas kartą baigus vieną iš užduočių, programinis kodas keliamas į bendrą repozitoriją ir peržiūrimas kitų komandos narių dėl galimų defektų. Produktą daug kartų ištestavus ir išsamiai dokumentavus, šis paruošiamas naudojimui. Jei per visą produkto gyvavimo laiką atsiranda kritinių klaidų, įmonė jas sutvarko savo lėšomis. Produktai yra prižiūrimi ir tobulinami visą laiko terminą, kuris su klientu buvo sutartas prieš pradėdant vystyti projektą.

## **2. Praktikos dalis**

### **2.1. Techninė užduotis**

#### **2.1.1. Sistemos apibrėžimas**

Kuriamas produktas – „Xplicity“ įmonės vidinė informacinė sistema, kurios pagrindinė paskirtis – kaupti duomenis apie darbuotojų atostogas, siekiant lengviau atlikti jų apskaitą. Tačiau tai nėra vienintelė sistemos funkcija. Galutinis produktas turėtų aprėpti ir kitokius administruojamus įmonės procesus, tokius kaip biurai priklausančių daiktų inventORIZacija, taip pat – apklausų pildymas. Praktiką atliekančios studentės darbo dalis – viršvalandžių posistemė, pirminiai testai.

#### **2.1.2. Bendras veiklos tikslas**

Sukūrus ir įdiegus sistemą bei pradėjus ją naudoti tikimasi pasiekti patogesnio ir efektyvesnio įmonės atostogų bei biurai priklausančių daiktų administravimo. Sistemoje išsaugoma informacija apie sukauptas atostogas, jų prašymus ir panaudojimą, įmonės inventorių, atliekami automatiniai skaičiavimai (viršvalandžių pavertimas į atostogų dienas, daiktų nuvertėjimas).

Praktiką atliekančios studentės darbo tikslas – realizuoti viršvalandžių posistemę, kuri leistų sukauptus viršvalandžius panaudoti prašant apmokamų atostogų dienų, taip pat parašyti pirminius testus jau egzistuojančioms sistemos funkcijoms.

#### **2.1.3. Sistemos pagrindumas**

Įmonė siekia sukurti sistemą, kuri būtų kiek įmanoma labiau pritaikyta jos darbuotojų individualiems poreikiams, palengvintų atostogų bei biurai priklausančių daiktų administravimą. Siekiama, kad darbuotojai galėtų internetu nesudėtingai pateikti atostogų prašymus, klientai ir administratoriai juos patvirtinti nuotoliniu būdu, o visa atostogų istorija būtų kaupiama duomenų bazėje ir prieinama norint atlikti apskaitą. Yra poreikis automatizuoti kuo daugiau vidinių procesų, kurie reikalauja administratoriaus bendravimo su darbuotoju, bet galimi atlikti savarankiškai ir per atstumą.

#### **2.1.4. Konkurencija rinkoje**

Kadangi produkto funkcijos yra pritaikytos konkrečiai įmonei ir jos poreikiams, sudėtinga atrasti jau egzistuojančią sistemą, kuri atliktų visas norimas užduotis.

Lietuvoje yra panašūs šie produktai:

- *SimpleApprove* – patogus personalo dokumentų valdymas [1]. Į Lietuvos rinką orientuotas produktas, siūlantis atostogų prašymų teikimą, pritaikytą skirtingiems atostogų tipams. Darbuotojas gauna pranešimą apie artėjančią atostogų pabaigą, visos atostogos bei neatvykimai į darbą matomi kalendoriuje, automatiškai generuojami dokumentai.  
Funkcijos, kurių trūksta: nėra inventORIZacijos ir apklausų pildymo posistemių; nėra galimybės darbuotojui skirti klientą, kuris taip pat patvirtintų atostogų prašymą; kalendorius nerodo darbuotojų gimtadienių bei sveikatos pažymos galiojimo pabaigos.
- *Elektroninės atostogos* – atostogų planavimo ir valdymo sistema online [2]. Į Lietuvos rinką orientuotas produktas, siūlantis savarankišką atostogų prašymų teikimą, jų istoriją, ataskaitų generavimą. Galima sekti darbuotojų atostogų dienų likutį, matyti bendrą atostogų kalendorių.  
Funkcijos, kurių trūksta: nėra inventORIZacijos posistemės, apklausų pildymo, informacijos apie darbuotojų gimtadienius bei sveikatos pažymos galiojimo pabaigą.

Kadangi yra planuojama galutinį įmonės produktą pristatyti Lietuvos ir visos Europos rinkoms, aktualūs ir užsienio konkurentai:

- *Calamari* – žmogiškųjų išteklių valdymas [3]. Galimybė registruoti atostogas, pasirinkti iš kelių skirtingų tipų, skirti darbuotoją, kuris atsakingas už konkretaus žmogaus prašymų tvirtinimą, matyti atostogų istoriją.

Funkcijos, kurių trūksta: nėra kalendoriaus, inventorizacijos ir apklausų pildymo posistemių.

- *cake.hr* – žmogiškųjų išteklių valdymas [4]. Galima teikti atostogų prašymus, matyti atostogų grafiką, pamainų kalendorių bei apmokamų atostogų likutį, generuoti ataskaitas, parinkti darbuotoją, kuris tvirtins konkrečių asmenų prašymus.

Funkcijos, kurių trūksta: nėra inventorizacijos ir apklausų pildymo posistemių; nėra galimybės darbuotojui skirti klientą, kuris taip pat patvirtintų atostogų prašymą; kalendorius nerodo darbuotojų gimtadienių bei sveikatos pažymos galiojimo pabaigos.

Kadangi „Xplicity“ įmonėje dirba iki 30 darbuotojų, žemiau pateikiama konkurentų mėnesinio mokesčio lentelė, paskaičiuota būtent trisdešimties žmonių kompanijai (1 lentelė).

Produkto pavadinimas	SimpleApprove	Elektroninės atostogos	Calamari	cake.hr
Kaina	40 €	60 €	100 \$	480 €

1 lentelė. Konkurentų mėnesinio mokesčio lentelė

### 2.1.5. Prototipai ir pagalbinių informacija

Šiai sistemai įmonė neturėjo ankstesnės versijos, tačiau jau kurį laiką buvo galvojama apie įgyvendinimą, išsamiai aprašomos ir detalizuojamos norimos funkcijos, vis konkrečiau apibrėžiama norima išvaizda. Iš pradžių buvo atliekamas tyrimas, ar rinkoje jau yra panašus produktas, kuris atitiktų visus reikalavimus. Radus panašius, bet nepilnai įmonės poreikius patenkinančius produktus, imtasi pradinės sistemos versijos kūrimo.

Pradinė sistema turėjo šias funkcijas:

- Prisijungimas prie administratoriaus ar paprasto darbuotojo paskyros (visos funkcijos, kurios priklauso darbuotojui, yra prieinamos ir administratoriui).
- Nesudėtingas ir mažafunkcis naujos darbuotojo paskyros sukūrimas.
- Galimybė paprašyti atostogų (atostogos trijų tipų: kasmetinės, tėvadienis/mamadienis, mokymosi).
- Galimybė gauti atostogų patvirtinimą (pateikus prašymą, išsiunčiama užklausa į visą sistemą administruojančių darbuotojų el. pašta).

### 2.1.6. Siūlomas sprendimas

Visos funkcijos, paminėtos ankstesniame skyrelyje, buvo plėtojamoms ir pildomos kuriant pagrindinę sistemą, nebuvo planuojama nieko atsisakyti.

- Pasiūlyta naudotis esama programos išvaizda tol, kol sistema įgis pagrindines funkcijas, o vėliau papildomai orientuotis į dizainą ir jį sutvarkyti atsižvelgiant į vartotojų poreikius ir pastebėjimus.
- Nuspręsta, kad pagrindinė sistema turi turėti kalendorių, filtruojamą pagal skirtingas komandas, kad būtų galima matyti vizualų atostogų grafiką ir šitaip lengviau sekti visus patvirtintus atostogų prašymus.
- Sistemai patikėti kai kurie skaičiavimai (viršvalandžių, inventoriaus nuvertėjimo), šitaip palengvinant įmonės buhalterinį darbą.
- Gerai apgalvota sistemos pranešimų/ataskaitų posistemė, siekiant vartotojui pateikti kuo daugiau informacijos apie įvykstančius būsenų pasikeitimus (pavyzdžiui, siunčiami pranešimai ir apie patvirtintus, ir apie atmestus atostogų prašymus; darbuotojui apleidus prašymą, bandydamas jį patvirtinti/atmesti, administratorius matys atitinkamą pranešimą).

### 2.1.7. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti

Darbo jėga, kuriant projektą, ne visą laiką išliko stabili, todėl sunku nustatyti, koks yra tikrasis tokios sistemos žmogiškųjų išteklių poreikis. Prie nuoseklaus sistemos kūrimo prisidėjo šeši praktikantai (keturi programų sistemų studentai, du informacinių technologijų studentai), tačiau bendrakūrėjų gretas papildydavo ir nauji programuotojai, norintys pagerinti savo žinias. Vieni, supažindinti su projektu ir pradėję darbą, neužilgo pasitraukė, todėl sunku nusakyti tikrąjį darbui reikalingų žmonių skaičių.

Pagal esamus rezultatus ir laiką, kuris buvo sugaištas jiems pasiekti, sistemai reikalingus žmogiškuosius išteklius galima skirstyti pagal žemiau pateiktą lentelę (2 lentelė). Joje pateiktas numanomas produkto (iki dabartinės versijos) įgyvendinimo laikas, darant prielaidą, kad sistemos kūrėjų skaičius nekinta (visada išlieka šeši pastovūs programuotojai), ir atsižvelgiant į skirtingas darbo valandas (nuo pilnos darbo dienos iki dešimties valandų per savaitę).

Darbo valandos	8 val. per dieną	4 val. per dieną	10 val. per savaitę
Sistemos įgyvendinimo laikas	3 mėnesiai	6 mėnesiai	12 mėnesių

2 lentelė. Numanomas sistemos įgyvendinimo laikas

Projektą baigiama įgyvendinti tik iš funkcionalumo pusės, darbas (nuo pradinės versijos iki dabartinio rezultato) iš viso truko apie šešis mėnesius. Iš viso teikiamos trys pagrindinės paslaugos:

- Atostogų prašymų pateikimas ir patvirtinimas nuotoliniu būdu, atostogų atvaizdavimas kalendoriuje, galimybė panaudoti sukauptus viršvalandžius.
- Įmonei priklausančių daiktų inventORIZACIJA ir nuvertėjimo automatinis skaičiavimas.
- Galimybė kurti apklausas, kurias gali užpildyti kiti sistemos vartotojai.

Sistema sudaryta iš aštuonių modulių, kurie susieti bendru funkcionalumu:

- Prisijungimo, paskyros ir vartotojo informacijos modulis.
- Pagrindinio (namų) puslapio bei meniu modulis.
- Darbuotojų modulis.
- Klientų modulis.
- Atostogų ir kalendoriaus modulis.
- Inventoriaus modulis.
- Apklausų modulis.
- Pranešimų modulis.

Bendras kodo eilučių skaičius: ~30 tūkst. programos kodo eilučių, ~4 tūkst. testų kodo eilučių.



## 2.2. Reikalavimų specifikacija

### 2.2.1. Komerčinė specifikacija

Projekto užsakovas yra UAB „Xplicity“, projekto vykdytojai – šeši žinias gilinantys programuotojai, vadovaujami ir padedami dviejų patyrusių įmonės darbuotojų. Taip pat prie projekto vystymo prisideda marketingo specialistė.

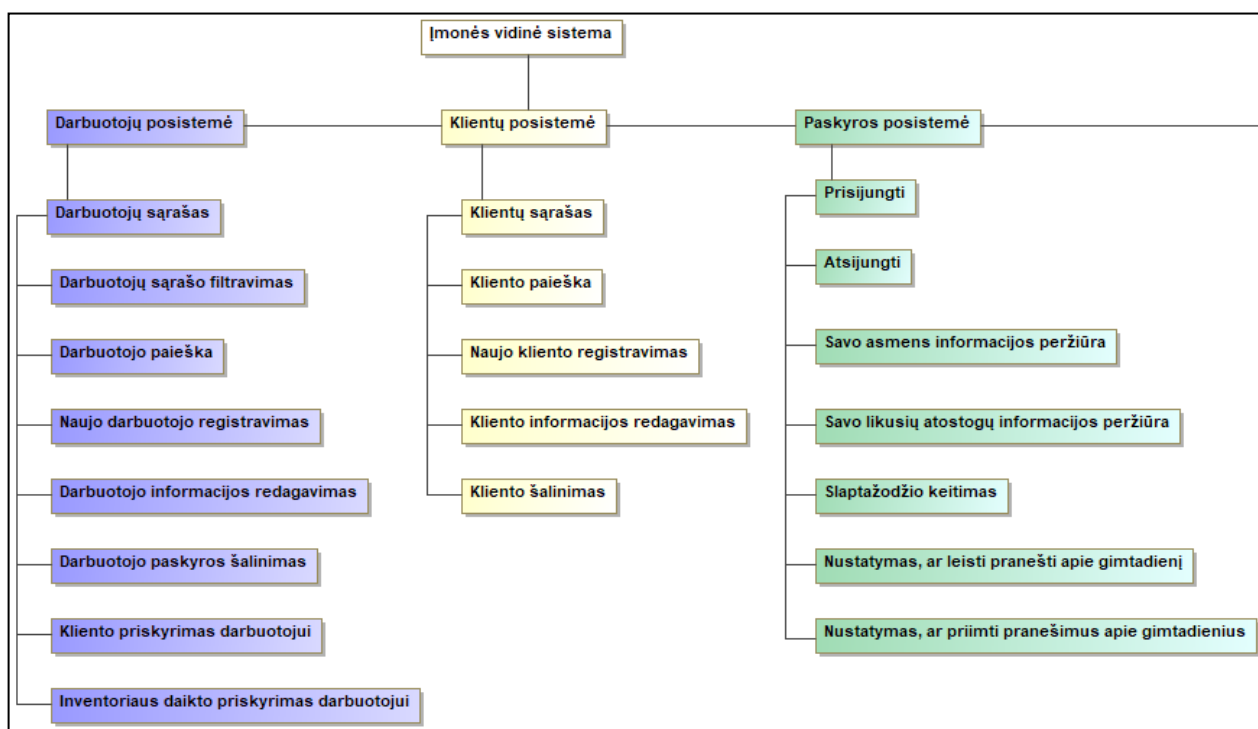
Produkto pradinis numatytas vartotojas yra „Xplicity“ įmonės darbuotojas (nuo programautojo, kuriam būtų patogiu nuotoliniu būdu pateikti atostogų prašymus ir pildyti apklausas, iki apskaitininko, kuriam dauguma sukaupų, apdorotų ir sugrupuotų duomenų stipriai palengvintų darbą). Vėliau produktą tikimasi pristatyti ir visos Europos rinkai, todėl marketingo specialistė vykdo apklausus, rengia pasiūlymus tam, kad produktas taptų kuo patrauklesnis ir kitoms suinteresuotoms Lietuvos bei užsienio šalių įmonėms.

Nors sistema ir gausiai pagelbėtų įmonės buhalteriniame darbe, suteiktų darbuotojams daugiau galimybių, sistemą sukurti nėra itin skubama. Labiausiai orientuojamasi į saugumą, lėtumą, bet kokią darbą, stengiamasi kuo daugiau testuoti, iš karto taisyti klaidas. Jokie laiko ribojimai nėra taikomi dėl darbo pobūdžio, kurį atlieka praktikantai, kuris, iš esmės, negali būti pilnai patikimas ir deramai suplanuotas.

Sistemos kūrimui yra naudojamos visos neapmokestintos ar įmonei jau priklausančios technologijos, darbo jėga taip pat nėra apmokama, o sudaryta iš savanoriškai dirbančių ir žinias gilinančių programuotojų, todėl produkto savikaina yra beveik minimali. Vieninteliai kaštai susidaro skaičiuojant valandas, kurias įmonės darbuotojai skiria vadovaudami bendram darbui.

### 2.2.2. Sistemos funkcijos

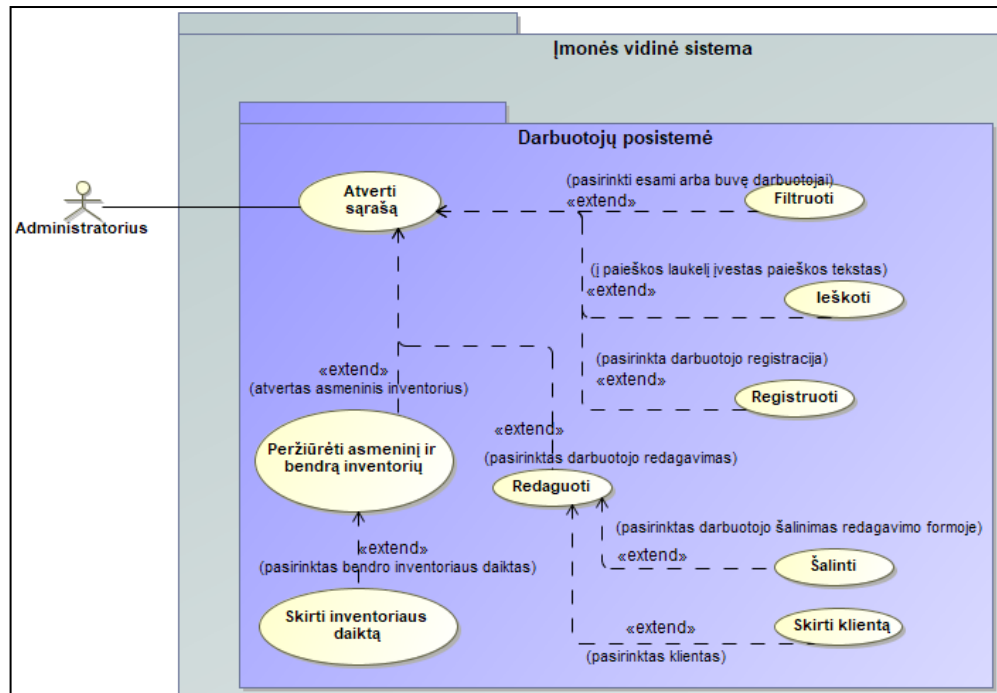
Šiame skyrelyje pateikiami visi sistemai keliami funkciniai reikalavimai. Sistema suskirstyta į septynias posistemas. Žemiau pateikiama pirmoji sistemos funkcijų hierarchijos dalis: darbuotojų, klientų ir paskyros posistemės (1 diagrama).



1 diagrama. Funkcijų hierarchija: darbuotojas, klientas, paskyra

Darbuotojų posistemė (2 diagrama):

- Naujo darbuotojo registravimas.
- Registruoto darbuotojo informacijos keitimas.
- Registruoto darbuotojo šalinimas.
- Galimybė darbuotojui skirti klientą (keli darbuotojai gali turėti tą patį klientą. Tai reiškia, kad darbuotojai priklauso vienai komandai ir dirba prie to paties užsakymo).
- Galimybė darbuotojui skirti daiktą iš įmonės inventoriaus (kol daiktas niekam nepriskirtas, šis priklauso biurui).

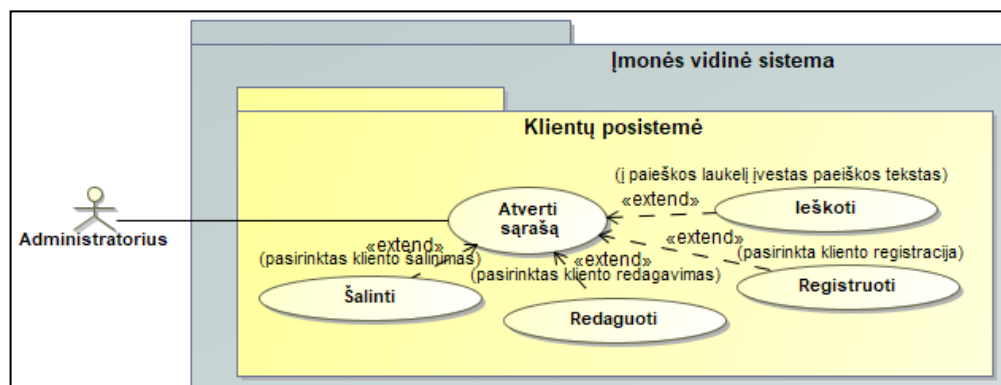


2 diagrama. Darbuotojų posistemės panaudojimo atvejai

Klientų posistemė (3 diagrama):

- Naujo kliento registravimas.
- Registruoto kliento informacijos keitimas.
- Registruoto kliento šalinimas.

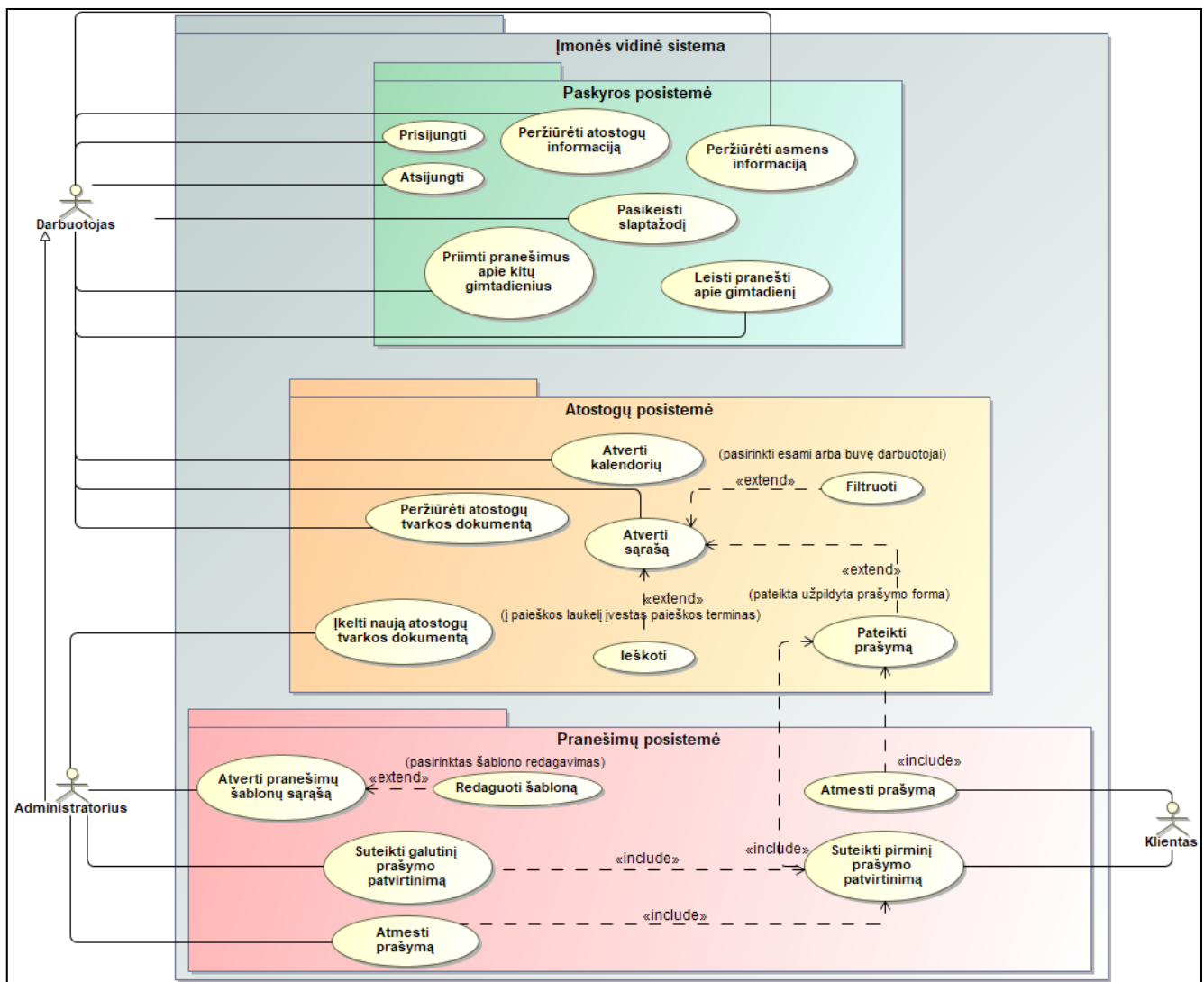
Klientas neturi prisijungimui skirtos paskyros. Yra registruojami tik kliento kompanijos pavadinimas, vadovo vardas ir pavardė bei duomenys, skirti susisiekimui (pvz, el. pašto adresas, kad būtų galima išsiųsti atostogų prašymą, kiti kontaktiniai duomenys).



3 diagrama. Klientų posistemės panaudojimo atvejai

Paskyros posistemė (4 diagrama):

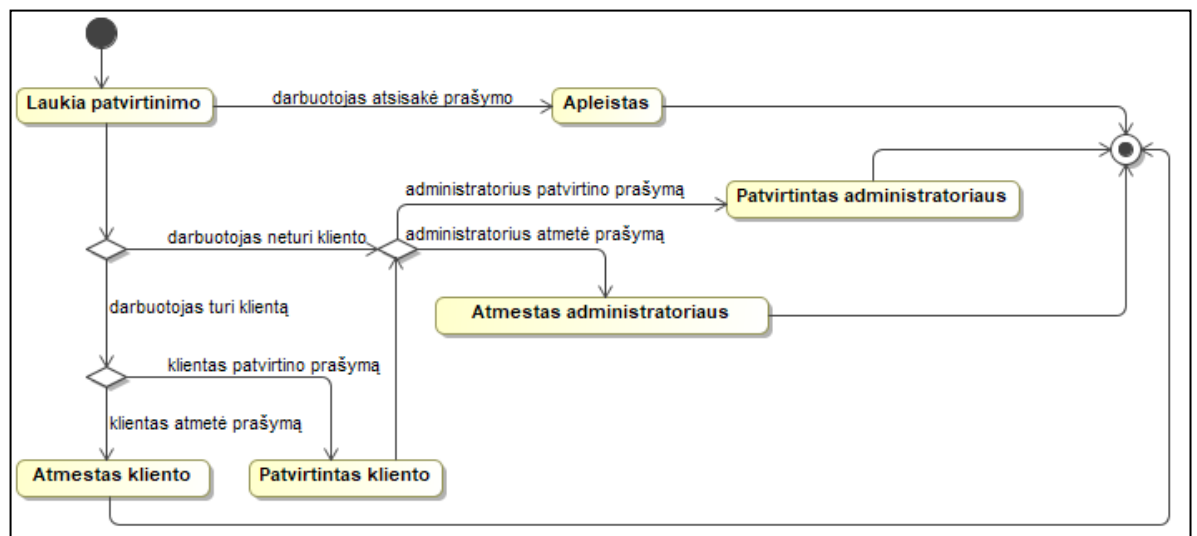
- Galimos dviejų tipų paskyros: administratoriaus ir darbuotojo. Administratorius paveldi visas darbuotojo funkcijas ir valdo kitas funkcijas, prie kurių įprastas darbuotojas neturi prieigos.
- Prisijungimas ir atsijungimas.
- Slaptažodžio keitimas.
- Savo asmeninės informacijos peržiūra, likusių kasmetinių atostogų bei sukaupytų viršvalandžių informacijos peržiūra.
- Galimybė savo gimtadienio informaciją padaryti vieša arba privačia (atėjus gimimo dienai, privatūs gimtadieniai kalendoriuje matomi tik administratoriui, apie viešą gimtadienį pranešama visiems darbuotojams pranešama el. laišku).



4 diagrama. Paskyros, atostogų ir pranešimų posistemių panaudojimo atvejai

Atostogų posistemė (4 diagrama):

- Visų atostogų prašymų pateikimas sąraše, kuriame galima matyti išsamią informaciją ir būsenų pasikeitimus. Galimos būsenos (5 diagrama):
  - laukia patvirtinimo;
  - atmestas kliento;
  - atmestas administratoriaus;
  - patvirtintas kliento;
  - patvirtintas administratoriaus;
  - apleistas.
- Visų galutinai patvirtintų (su būsena „patvirtintas administratoriaus“) prašymų vaizdavimas kalendoriuje.
- Naujo prašymo pateikimas.
- Galimybė atsisakyti prašymo, kol dar nepatvirtino nei administratorius, nei klientas (esant būsenoje „laukia patvirtinimo“).



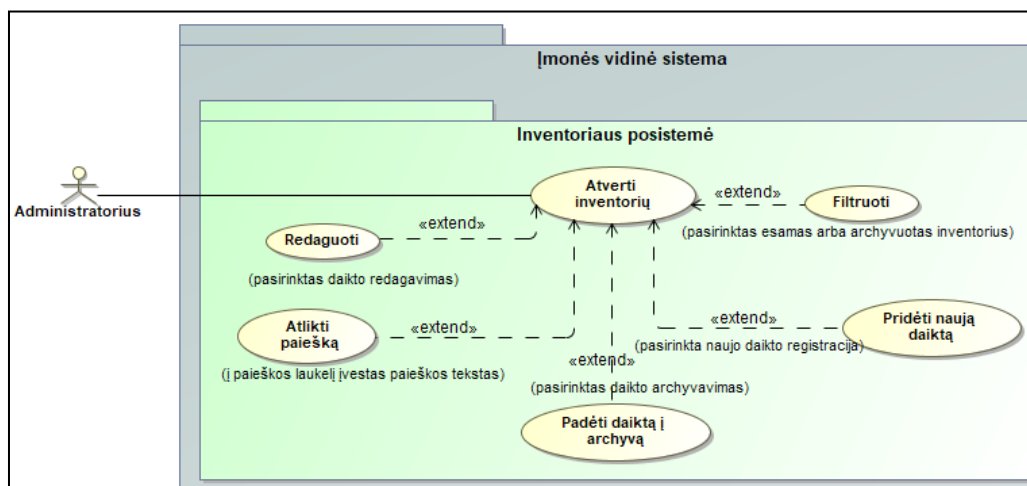
5 diagrama. Atostogų prašymo būsenų diagrama

Pranešimų posistemė (4 diagrama):

- Pranešimų šablonų redagavimas.
- Prašymo suteikti atostogas siuntimas el. paštu (pirma klientui, jei darbuotojas tokį turi, po to – administratoriui). Pridedama informacija apie norimus panaudoti viršvalandžius.
- Darbuotojo informavimas el. laišku apie patvirtintas arba atmestas atostogas.
- Automatiškai užpildytų ir spausdinimui paruoštų prašymo ir įsakymo dokumentų siuntimas el. paštu.
- El. laiškas mėnesio pabaigoje apie visas tą mėnesį suteiktas atostogas.
- El. laiškas visiems darbuotojams apie sekančią darbo dieną atostogausiančius kolegas.
- El. laiškas apie darbuotojo gimtadienį (tik tuo atveju jei darbuotojo gimtadienis viešas).

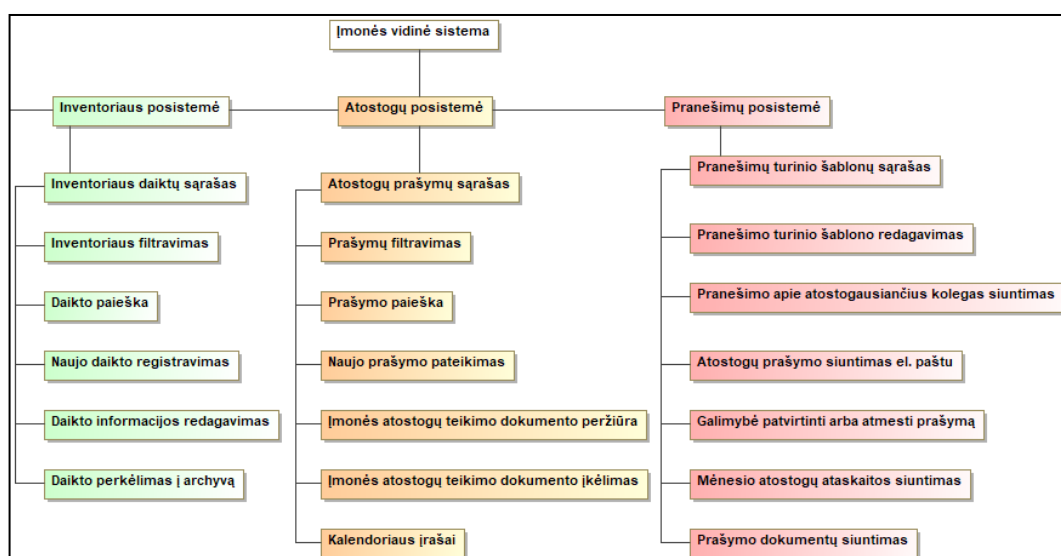
Inventoriaus posistemė (6 diagrama):

- Įmonės įsigyto daikto užregistravimas inventoriuje.
- Daikto informacijos redagavimas.
- Kasdienis daikto nuvertėjimas.
- Pilnai nuvertėjusių daiktų perkėlimas į archyvą.



6 diagrama. Inventoriaus posistemės panaudojimo atvejai

Žemiau pateikiama likusi sistemos funkcijų hierarchija: inventoriaus, atostogų ir pranešimų posistemės (7 diagrama):



7 diagrama. Funcijų hierarchija: inventorius, atostogos, pranešimai

### 2.2.3. Apribojimai

Šiame skyrelyje pateikiami sistemos kūrimo laiko bei sprendimo apribojimai, numatyta darbo vietos aplinka.

**Laikas.** Jokie laiko ribojimai nėra taikomi dėl darbo pobūdžio, kurį atlieka praktikantai. Šis negali būti pilnai patikimas ir deramai suplanuotas, todėl sunku nusakyti, kiek laiko užtruks pilnas sistemos įgyvendinimas.

**Darbo vieta.** Nėra numatyta specifinė darbo vietos aplinka. Sistema turi suteikti prieigą bet kuriam darbuotojui įvairiu paros metu iš darbo, namų ar kitos vietos prisijungti ir atlikti norimas užklausas.

**Apribojimai sprendimui:** stengtis naudoti numatytas pasirinktų technologijų funkcijas ir bibliotekas, integruoti kuo mažiau svetimų produktų. Tiesa, taip būtų galima išvengti kai kurių sudėtingų sprendimų ir pagreitinoti/palengvinti darbą esamu momentu, tačiau nėra garantijos, kad darbas nebus

apsunkintas vėliau, jei panaudoti trečios šalies komponentai nebus toliau reguliariai atnaujinami kartu su pasirinktu karkasu.

## 2.2.4. Duomenų ir rezultatų specifikacija

Šiame skyrelyje pateikiami skirtingų įrašų duomenys ir rezultatai. Įvedimo momentas grupuojamas pagal naujų įrašų sukūrimą ir esamų įrašų redagavimą (3 lentelė).

Darbuotojo modelis			
Pateikimas (GET)	Pavyzdys	Sukūrimas (POST)	Redagavimas (PUT)
<code>int</code> Id <code>string</code> Name <code>string</code> Surname <code>int</code> ClientId <code>DateTime</code> WorksFromDate <code>DateTime</code> BirthdayDate <code>int</code> DaysOfVacation <code>double</code> FreeWorkDays <code>double</code> OvertimeHours <code>double</code> OvertimeDays <code>int</code> ParentalLeaveLimit <code>int</code> CurrentAvailableLeaves <code>int</code> NextMonthAvailableLeaves <code>string</code> Email <code>string</code> Role <code>string</code> Position <code>DateTime</code> HealthCheckDate <code>EmployeeStatusEnum</code> Status	1 Vardenis Pav 1 2020-01-01 2020-01-01 20 15 8 2 0 0 0 em@email.com Admin Biuro vadovė 2020-01-01 0 (Current)	<code>string</code> Name <code>string</code> Surname <code>int</code> ClientId <code>DateTime</code> WorksFromDate <code>DateTime</code> BirthdayDate <code>int</code> DaysOfVacation <code>int</code> ParentalLeaveLimit <code>double</code> FreeWorkDays <code>string</code> Email <code>string</code> Password <code>string</code> Role <code>String</code> Position <code>bool</code> IsManualHolidaysInput <code>DateTime</code> HealthCheckDate <code>EmployeeStatusEnum</code> Status	<code>string</code> Name <code>string</code> Surname <code>int</code> ClientId <code>DateTime</code> WorksFromDate <code>DateTime</code> BirthdayDate <code>int</code> DaysOfVacation <code>int</code> ParentalLeaveLimit <code>int</code> OvertimeHours <code>double</code> FreeWorkDays <code>string</code> Email <code>string</code> Password <code>string</code> Role <code>string</code> Position <code>DateTime</code> HealthCheckDate <code>EmployeeStatusEnum</code> Status
Kliento modelis			
Pateikimas (GET)	Pavyzdys	Sukūrimas (POST)	Redagavimas (PUT)
<code>int</code> Id <code>string</code> CompanyName <code>string</code> OwnerName <code>string</code> OwnerSurname <code>string</code> OwnerEmail <code>string</code> OwnerPhone	1 Pavadinimas Vardenis Pavardenis cl@email.com 861234567	<code>string</code> CompanyName <code>string</code> OwnerName <code>string</code> OwnerSurname <code>string</code> OwnerEmail <code>string</code> OwnerPhone	<code>string</code> CompanyName <code>string</code> OwnerName <code>string</code> OwnerSurname <code>string</code> OwnerEmail <code>string</code> OwnerPhone
Atostogų prašymo modelis			
Pateikimas (GET)	Pavyzdys	Sukūrimas (POST)	Redagavimas (PUT)
<code>int</code> Id <code>int</code> EmployeeId <code>string</code> EmployeeFullName <code>HolidayType</code> Type <code>DateTime</code> FromInclusive <code>DateTime</code> ToInclusive <code>int</code> OvertimeDays <code>double</code> OvertimeHours <code>HolidayStatus</code> Status <code>DateTime</code> RequestCreatedDate <code>int</code> ConfirmerAdminId <code>int</code> ConfirmerClientId <code>string</code> ConfirmerFullName <code>string</code> RejectionReason	1 2 Vard Pava 0 (Annual) 2020-01-02 2020-01-05 1 4 0 (Pending) 2020-01-01 1 1 Vardenis Pav Rejection reason text	<code>int</code> EmployeeId <code>HolidayType</code> Type <code>DateTime</code> FromInclusive <code>DateTime</code> ToInclusive <code>int</code> OvertimeDays <code>HolidayStatus</code> Status <code>DateTime</code> RequestCreatedDate <code>int</code> ConfirmerAdminId <code>int</code> ConfirmerClientId <code>string</code> ConfirmerFullName <code>string</code> RejectionReason	<code>int</code> EmployeeId <code>HolidayType</code> Type <code>DateTime</code> FromInclusive <code>DateTime</code> ToInclusive <code>int</code> OvertimeDays <code>HolidayStatus</code> Status <code>DateTime</code> RequestCreatedDate <code>int</code> ConfirmerAdminId <code>int</code> ConfirmerClientId <code>string</code> ConfirmerFullName <code>string</code> RejectionReason

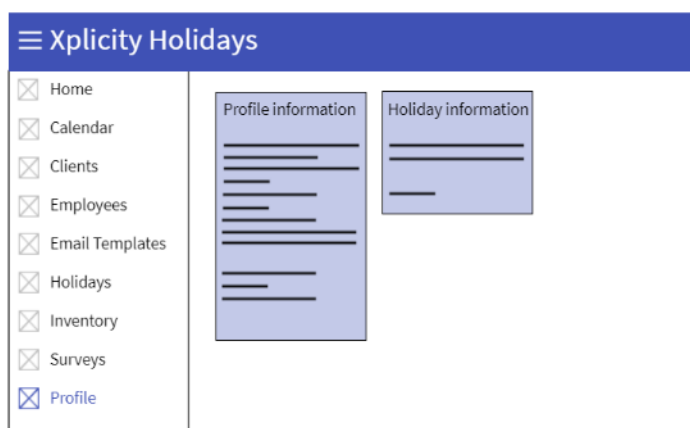
Inventoriaus daikto modelis			
Pateikimas (GET)	Pavyzdys	Sukūrimas (POST)	Redagavimas (PUT)
<code>int</code> Id <code>string</code> Name <code>string</code> SerialNumber <code>DateTime</code> PurchaseDate <code>DateTime</code> ExpiryDate <code>string</code> Comment <code>int</code> InventoryCategoryId <code>InventoryCategory</code> Category <code>decimal</code> OriginalPrice <code>int</code> EmployeeId <code>string</code> AssignedTo <code>bool</code> Archived <code>decimal</code> CurrentPrice	1 Laptop L-123456 2020-05-01 2023-05-01 Comment text 1 0 (Computer) 888.5 1 Vardenis Pav false 888.48	<code>string</code> Name <code>string</code> SerialNumber <code>DateTime</code> PurchaseDate <code>DateTime</code> ExpiryDate <code>string</code> Comment <code>int</code> InventoryCategoryId <code>decimal</code> OriginalPrice <code>int</code> EmployeeId	<code>string</code> Name <code>string</code> SerialNumber <code>DateTime</code> PurchaseDate <code>DateTime</code> ExpiryDate <code>string</code> Comment <code>int</code> InventoryCategoryId <code>decimal</code> OriginalPrice <code>int</code> EmployeeId <code>bool</code> Archived
Pranešimų šablono modelis			
Pateikimas (GET)	Sukūrimas (POST)		Redagavimas (PUT)
<code>int</code> Id <code>string</code> Purpose <code>string</code> Subject <code>string</code> Template <code>string</code> Instructions	<code>string</code> Purpose <code>string</code> Subject <code>string</code> Template <code>string</code> Instructions		<code>string</code> Purpose <code>string</code> Subject <code>string</code> Template <code>string</code> Instructions
Pavyzdys: "Purpose": "Client Confirmation", "Subject": "One of your employees is requesting confirmation for holidays", "Template": "Hello, {client.name},\n\nAn employee {employee.fullName} is intending to go on {holiday.type} holidays from {holiday.from} to {holiday.to} (inclusive).\n\nClick this link to confirm or decline: {holiday.confirm}", "Instructions": "{client.name} - Receiving client's first name.\n {employee.fullName} - Employee's full name.\n{holiday.type} - Holiday's type.\n{holiday.from} - Holiday's starting date.\n{holiday.to} - Holiday's ending date.\n{holiday.confirm} - Holiday's confirmation or rejection link."			

3 lentelė. Duomenų ir rezultatų specifikacija

## 2.2.5. Vartotojo sąsajos specifikacija

Šiame skyrelyje aprašomi visi vartotojo sąsajai keliami reikalavimai, pateikiami eskizai (užsakovas eskizų nepateikė, jie nubraižyti pagal keliamus bendrus reikalavimus).

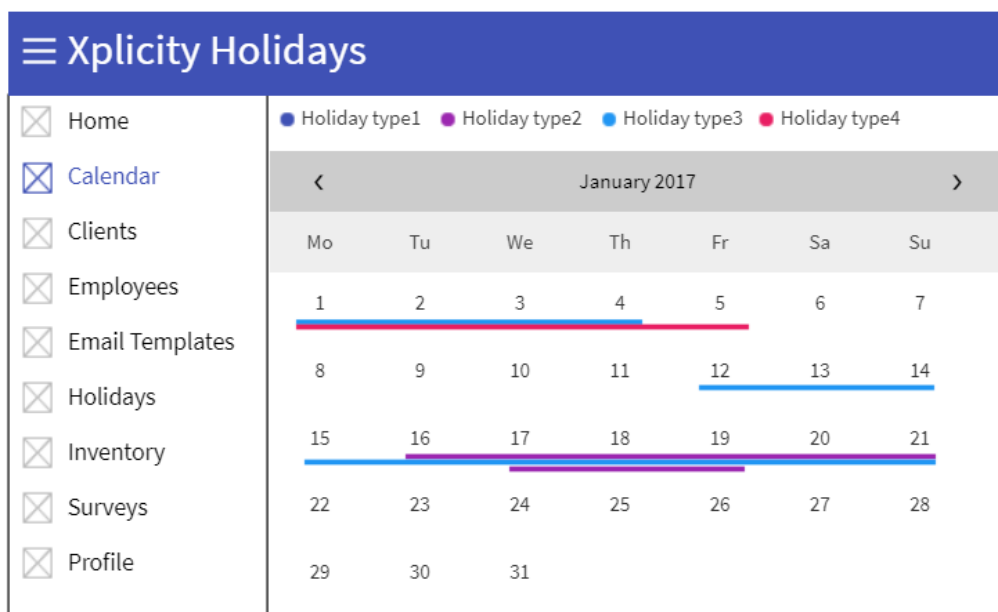
Sistemos išvaizda turi būti neapsunkinta ir minimali, funkcijos sugrupuotos ir pateikiamos vienoje vietoje pagal savo pobūdį. Pagrindiniame meniu turi būti ne daugiau kaip dešimt skyrių, kiekvienas turi turėti aiškų pavadinimą ir jį atitinkančią ikoną (1 pav.).



1 pav. Sistemos meniu ir vartotojo profilio eskizas

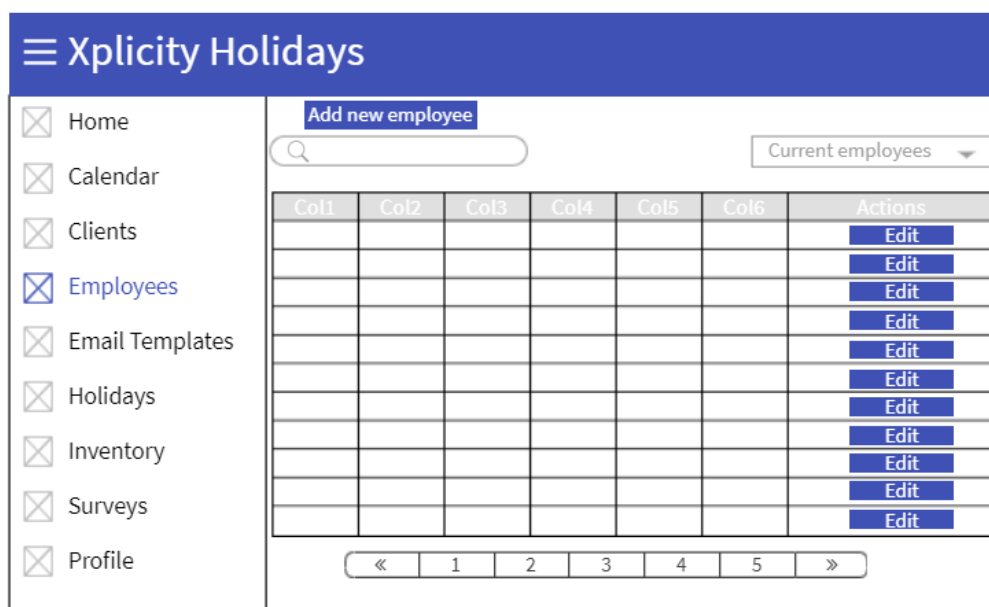


Kalendoriuje skirtingo tipo įvykiai (atostogos, gimtadienis, medicininės pažymos galiojimo pabaiga) turi būti pažymėti skirtingomis spalvomis. Virš kalendoriaus turi būti matoma kiekvienam tipui skirta spalva; įvykiai žymimi juostomis, besitęsiančiomis per visą kalendoriaus dienos langelį (2 pav.).



2 pav. Kalendoriaus eskizas

Duomenų lentelėse rodoma tik pati svarbiausia informacija (norint pamatyti daugiau duomenų apie įrašą, tai galima padaryti paspaudus „Edit“ (įrašo redagavimo) mygtuką). Rečiau naudojamos funkcijos turi būti daug mažiau pastebimos nei dažnai naudojamos funkcijos: norint ištrinti įrašą, tai galima padaryti atvėrus redagavimo formą; filtravimo ir paieškos funkcijų pasirinkimai blankesnės spalvos nei redagavimo ar naujo įrašo mygtukai (3 pav.).



3 pav. Sistemos duomenų lentelės eskizas



## 2.2.6. Realizacijai keliami reikalavimai

Šiame skyrelyje aprašomi visi realizacijai keliami reikalavimai.

Reikalavimai saugumui:

- Elektroniniu paštu siunčiamos prašymų patvirtinimo nuorodos, dokumentų atsissiuntimo nuorodos. Nuorofoje neturi būti jokių darbuotojo, atostogų prašymo ar dokumento duomenų (pavadinimų, indentifikatorių), kurie, atlikus nesudėtingus pakeitimus, suteiktų prieigą prie kitų duomenų. Visi identifikatoriai turi būti laikomi atskiroje duomenų bazės lentelėje ir su nuoroda susieti tik pagal užšifruotą kodą.
- Pasikeitus savo slaptažodį (ar jį pakeitus administratoriui), senasis slaptažodis automatiškai ištrinamas ir įsigali naujas. Senasis saugumo žetonas turi baigti galioti, prisijungęs vartotojas turi būti atjungiamas nuo sistemos.

Reikalavimai veikimo sąlygoms:

- Sistema turi būti prieinama visiems įmonės darbuotojams bet kuriuo paros metu ir savaitgaliais, ir darbo dienomis.
- Sistema turi kas dieną numatyti laiku atlikti fonines operacijas: už darbo dieną priskaičiuoti atitinkamą kiekį atostogų, biurui priklausantiems daiktams paskaičiuoti nuvertėjimą, išsiųsti atitinkamus laiškus. Visos šios įvykdytos operacijos turi būti registruojamos, kad, sistemai nustojus veikti, būtų įmanoma sekti, ar pavyko sėkmingai atlikti skaičiavimus, išsiųsti laiškus ir pan.

Teisiniai reikalavimai:

- Svarbu itin atidžiai vystyti funkcijas, kurios gali pažeisti darbuotojų privatumą. Pavyzdžiui, negali būti atsitiktinai išplatinta konfidenciali informacija, kontaktai ar duomenys, kuriuos darbuotojas nustatė matomus administratoriams, bet neprieinamus kitiems (kaip gimimo data, atostogų prašymai).
- Visada gali įsivelti žmogiškų klaidų. Audito žurnale turi būti registruojami visi sistemos vartotojų atliekami pakeitimai tam, kad būtų įmanoma peržiūrėti veiksmų istoriją ir atsekti, kur įsivėlė klaida.

Reikalavimai sistemos priežiūrai:

- Sistema turi būti intensyviai prižiūrima bent pirmaisiais paleidimo mėnesiais, kai susijungs ir pradės naudotis numatyti galutiniai vartotojai. Testiniai duomenys bus pakeisti tikrais, žymiai padidės apkrova.
- Turi būti įpareigoti keli asmenys, kurie reguliariai peržiūri audito žurnalą ir patikrina, kad viskas tinkamai veikia, atliekamos visos foninės operacijos.

Galutinis produktas turės nuosavą domeną (app.xplicity.com), bus pritaikytas mobiliam aplinkai, todėl reikės į tai atsižvelgiant

- atlikti vartotojo sąsajos ir dizaino pakeitimus,
- nustatyti, kurios sistemos sritys tinka/netinka mobiliam aplinkai,
- įgalinti Progressive-Web-App konfigūraciją.

### 2.2.7. Techninė specifikacija

Šiame skyrelyje pateikiama visos technologijos, reikalingos sistemos realizacijai, bei sąrašas programų, kurios reikalingos, norint naudotis sistema.

Prieš pradėdant sistemos kūrimo darbus rekomenduojama parsisiųsti ir įsidiegti šią programinę įrangą:

- Node.js (v12+),
- Visual Studio Code,
- Visual Studio 2019 IDE,
- Microsoft SQL Server Management Studio 18,
- Sourcetree vartotojo sąsajai Git'ui.

Sistemai sukurti buvo pasirinktos šios technologijos:

- **ASP.NET Core, versija 3.1.** Tai C# programavimo kalbos karkasas, yra atviro kodo, diegimo procesas itin lengvas, tie patys komponentai gali atlikti visas savo funkcijas tiek debesų platformoje, tiek kliento telefone ar kompiuteryje [5].  
Darbui reikalinga: **Visual Studio 2019 IDE.**
- **Angular, versija 9.** Tai TypeScript programavimo kalba paremtas atviro kodo karkasas. Yra patikimas, nes palaikomas Google Angular komandos bei kitų korporacijų. Integruotas su CLI (komandinių eilučių programa, kuri automatizuoja karkaso komponentų kūrimą) [6].  
Darbui reikalinga: **Node.js, Visual Studio Code.**
- **Microsoft SQL Server.** Tai duomenų bazių, skirtų ne itin didelės apimties programoms, valdymo sistema. Ši duombazė yra nemokama ir suderinama su „Microsoft Azure SQL database“ [7].  
Darbui reikalinga: **Microsoft SQL Server Management Studio 18.**
- **Microsoft Azure.** Tai debesų paslauga, skirta automatizuoti programų kompiliavimą ir testavimą, taip pat palaiko veiklos organizavimą lanksčiais (Agile) metodais. Azure aplinkoje saugomas suplanuotų sistemos funkcijų sąrašas, kuris nuolat papildomas ir detalizuojamas [8].  
Darbui užtenka paprastos interneto naršyklės.

Visa programos išvaizda laikoma atskirame ClientApp aplanke, vystoma naudojantis Angular (HTML, CSS ir typescript kalbomis).

Visa programos logika, skaičiavimai ir kontrolieriai laikomi atskirame .sln projekte, vystomi naudojantis ASP.NET Core, su C# kalba.

Norint naudotis galutine sistemos versija bet kuriam darbuotojui užteks turėti mobilųjį įrenginį ar kompiuterį ir paprastą interneto naršyklę.

## **2.3. Projektavimo eiga ir sistemos projektas**

### **2.3.1. Projektavimo valdymas ir eiga**

Prieš pradedant sistemos realizaciją nebuvo vykdomi jokie pasiruošimo darbai, kurie apimtų infrastruktūros, sprendimų ar optimizavimo vizualizaciją diagramomis. Projektavimui skirta UML modeliavimo kalba ir jai specializuoti įrankiai nebuvo naudojami. Tačiau programinės įrangos kūrimo inžinerinė veikla beveik visada apima UML modeliavimą, todėl visos diagramos savarankiškai nubraižytos praktiškai atliekančios studentės, siekiant apie projekto struktūrą suteikti daugiau vizualios informacijos.

Kadangi sistemos užsakovas yra pati įmonė, vykdomi dažni susitikimai ir vaizdo konferencijos, kurių metu pildomas ir keičiamas reikalavimų sąrašas. Naujoms funkcijoms braižomos tik neformalios trumpalaikės diagramos, kol bendromis pastangomis yra sugalvojamas sprendimas ir su menkiausiomis detalėmis perkeliamas į laukiančių darbų sąrašą. Nuo pradinės sistemos versijos architektūriniai sprendimai nebuvo stipriai keičiami, siekiant išlaikyti vientisumą skatinama naujas funkcijas įgyvendinti remiantis jau egzistuojančia realizacija.

### **2.3.2. Projektavimo technologija**

Nors sistema nėra sudėtinga, norint užtikrinti jos išliekamąją vertę ir palaikymą ją plečiant, taisant aptiktus defektus, architektūros dokumentavimas yra reikalingas ir svarbus.

Projektuojant sistemos realizaciją panaudota UML (Unified Modeling Language) modeliavimo kalba. Ji yra vizuali, leidžianti aprašyti pačius įvairiausius projektavimo sprendimus grafiškai. UML modeliai padeda daug lengviau perprasti sistemos veikimą ir struktūrą, todėl naudojami dokumentuojant architektūrą, aptariant naujų funkcijų realizaciją ar defektų taisymą.

Pasirintas vienas žinomiausių ir labiausiai pripažintų UML įrankių programinės įrangos rinkoje – „MagicDraw“. Sukurtas ir vystomas „No Magic“ kompanijos, šis įrankis visame pasaulyje yra sulaukęs daug pripažinimo ir apdovanojimų dėl savo siūlomų patogių ir itin efektyvių priemonių kuriant projektavimo modelius. Juo naudojasi verslo ir programų analitikai, programuotojai. Lankstus ir įvairiapusis įrankis tinka visoms sistemoms ir duomenų bazėms, kurios orientuotos į objektinį programavimą [9].

### **2.3.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos**

Šiame skyrelyje pateikiama informacija apie realizacijai tinkamas operacines sistemas, automatizavimo priemones darbui palengvinti bei visas sistemai naudojamas programavimo kalbas. Kai kurių žemiau išvardintų technologijų išsamų aprašymą galima rasti „[Techninės specifikacijos](#)“ skyrelyje.

Sistemos vartotojo sąsaja programuojama HTML, CSS ir Typescript kalbomis, viską apjungia Angular karkasas. Mikroservisų programavimui naudojamas .NET Core karkasas, programos tekstas rašomas C# kalba.

Darbas automatizuotas ir palengvintas naudojantis integruotomis programavimo aplinkomis: Visual Studio 2019 IDE, Visual Studio Code, JetBrains Ultimate. Pastarasis paketas suteikia darbą itin palengvinančius Visual Studio plėtinius:

- ReSharper – produktyvumo įrankis, papildantis Visual Studio įvairiomis kodo redagavimo, taisymo ir navigavimo funkcijomis.
- dotCover – kodo testinio padengimo analizės įrankis, pritaikytas vienetų testams ir suteikiantis grafinę analizės rezultatų vizualizaciją.

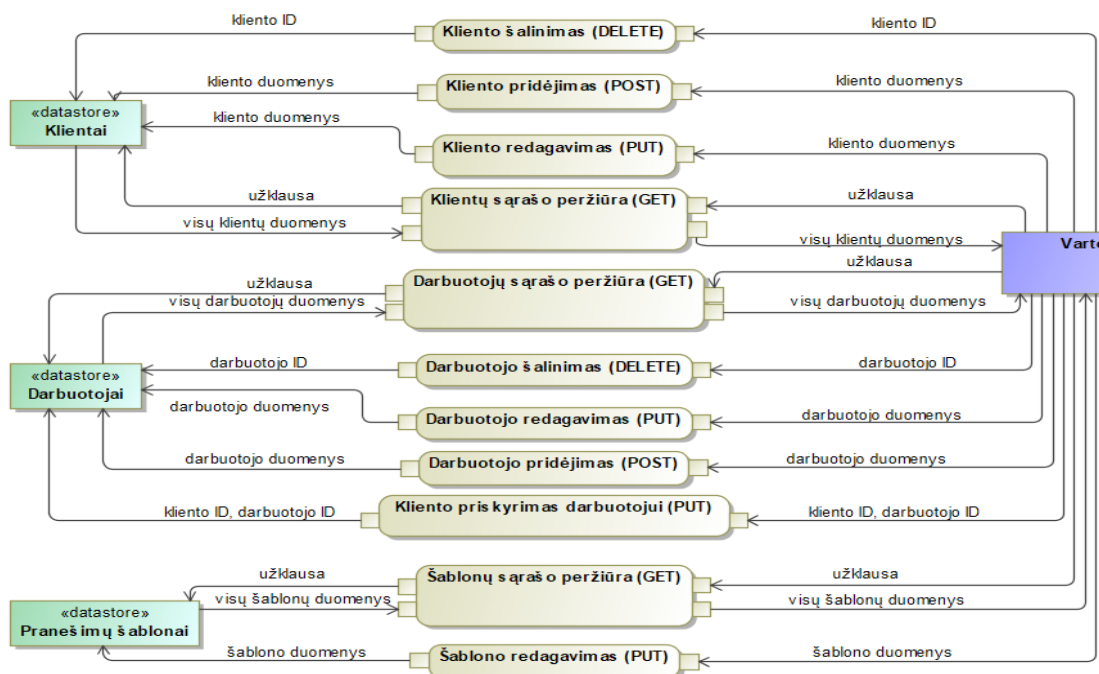
Taip pat naudojamas versijavimo kontrolės įrankis Git ir jam skirta grafinė sąsaja „Sourcetree“ (galima bet kuri kita grafinė sąsaja, jeigu ji programuotojui atrodo patogesnė). Šie įrankiai reikalingi tuo atveju, jeigu planuojama sąveikauti su esama programos repozitorija, norint greitos prieigos prie ankstyviausių atnaujinimų, ar siekiant perkelti bet kokius programos pakeitimus iš vietinės kopijos savo kompiuteryje į galutinio produkto repozitoriją. Toks įrankis itin pravartus:

- dirbant komandoje ir pakaitomis vystant tą patį projektą;
- norint pastoviai atnaujinti savo darbą ir galutinės versijos kopiją saugoti internete.

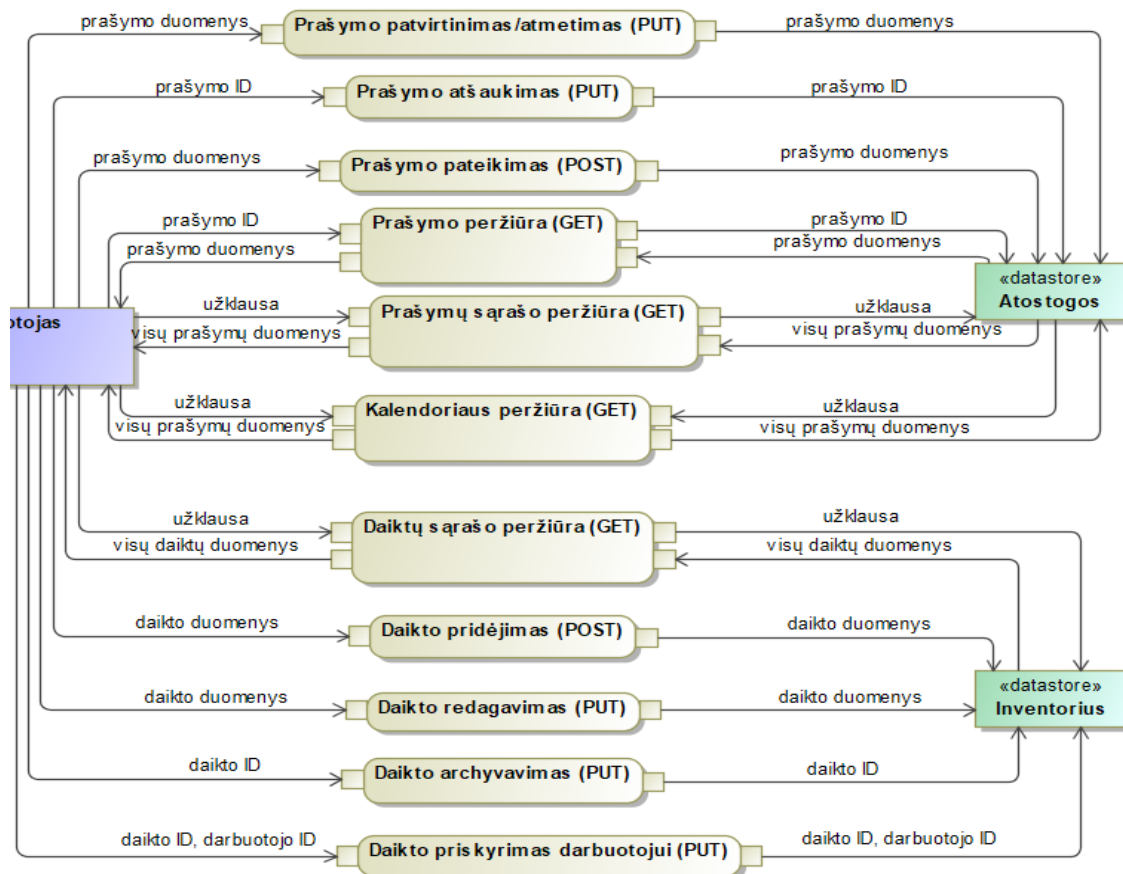
Sistema buvo kuriama ant Windows 10 operacinės sistemos, tačiau galima naudoti bet kurią OS, kuri palaiko aukščiau išvardintą programinę įrangą.

## 2.3.4. Informacijos srautai

Šiame skyrelyje pateikiama informacijos srautų vizualizacija nuo vartotojo iki duomenų bazės modelio (žr. 8 diagrama ir 9 diagrama). Diagramos braižymui MagicDraw įrankyje sukurta veiklos diagrama, duomenų srautų vaizdavimui pritaikyta panaudojant *Object Node*, *Data Store Node*, *Action* elementus ir *Object Flow* rodyklės tipą.



8 diagrama. Klientų, darbuotojų ir pranešimų šablonų posistemių informacijos srautai



9 diagrama. Atostogų ir inventoriaus posistemių informacijos srautai

Rodyklės nurodo duomenų judėjimo kryptį (pavyzdžiui, iš vartotojo išeinančios rodyklės rodo vartotojo įvedamus duomenis, kurie siunčiami kartu su užklausa; įeinančios rodyklės rodo gaunamus duomenis). Galimos užklausa pavadintos tais pačiais pavadinimais kaip ir vartotojo sąsajoje, galima matyti kokie duomenys siunčiami ir gaunami atliekant tam tikrus veiksmus. Taip pat pridėti naudojami kontrolerio metodai (POST – sukuriama naujas įrašas, PUT – redaguojamas esamas įrašas, GET – grąžinami įrašo/-ų duomenys, DELETE – ištrinamas esamas įrašas).

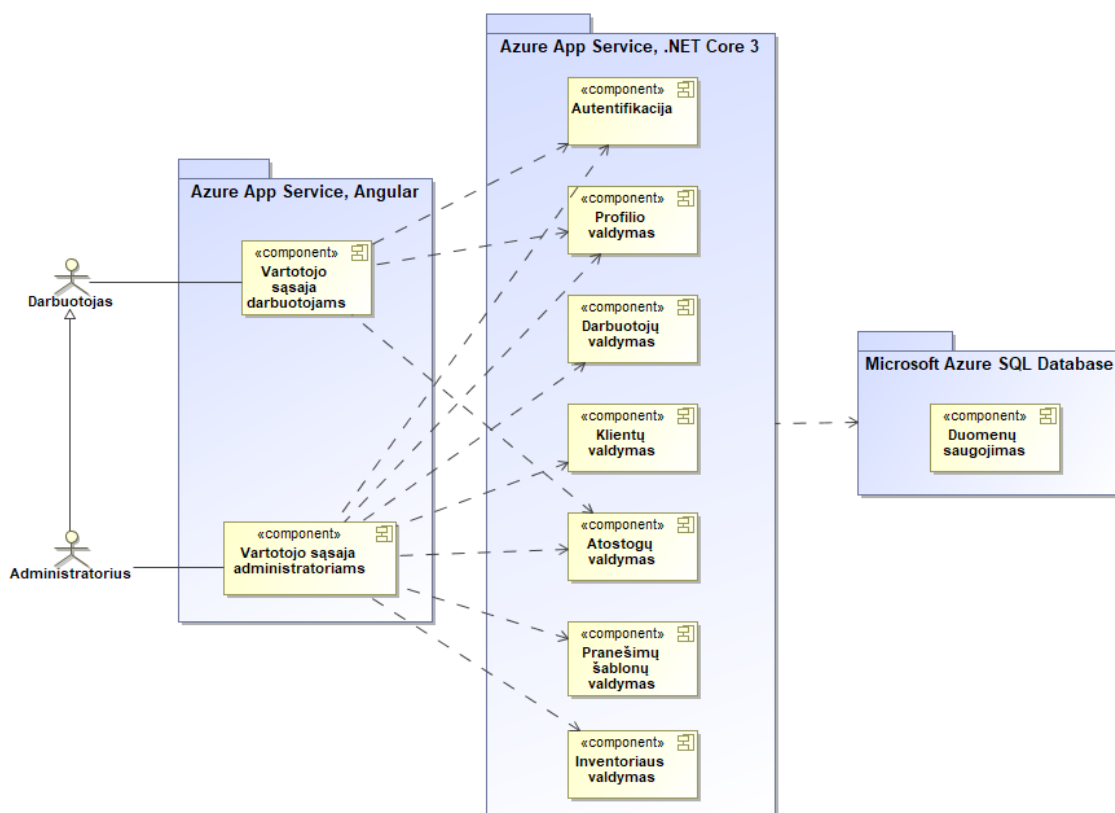
### 2.3.5. Sistemos sudėtis

Sistema yra MVC (Model-View-Controller) tipo, todėl jos sudedamosios dalys yra:

- Vartotojo sąsaja – puslapiai su interaktyviais vaizdiniais elementais, su kuriais sąveikaudamas vartotojas gali vykdyti įvairias užklausas.
- Kontroleriai. Užklausa su siunčiamais duomenimis pasiekia kontrolerio metodus. Kiekvienas kontroleris yra surištas su atitinkamu modeliu (kliento, darbuotojo, atostogos prašymo ir panašiai).
  - Jeigu užklausa sudėtinga ir reikalingi papildomi veiksmai, kontroleris kreipiasi į kitą lygmenį - mikroservisą.
  - Į infrastruktūrą įeina statiniai failai, kuriuose saugomos nekintančios reikšmės (teksto eilutės el. laišku, pritaikytos skirtingiems numatytiems atvejams ir pan.), pagalbiniai failai su keliems servisams pritaikomų metodų (laiko įrankiai: keleto metų pavertimas dienomis, sekančios darbo dienos datos radimas ir panašiai).
- Modelio įrašai, saugomi susietoje duomenų bazėje. Ir kontroleris, ir mikroservisas turi prieigą prie modelio repozitorijos metodų, kurie tiesiogiai valdo modelio įrašus, esančius susietoje duomenų bazėje.

### 2.3.6. Komponentai

Šiame skyrelyje pateikiama informacija apie sistemos komponentus ir jų pasiekiamumą skirtingiems prisijungusiems vartotojams (10 diagrama). Iš viso yra septyni mikroservisų komponentai, įskaičiuojant ir autentifikacijos mikroservisą. Visi jie naudojami viena Microsoft Azure SQL duomenų baze.



10 diagrama. Sistemos komponentai

### 2.3.7. Sąsajos įvertinimas pagal vartotojo patyrimą, profesinę terminologiją

Šiame skyrelyje pateikiamas vartotojo patyrimas lankantis sistemos internetinėje svetainėje ir atliekant įvairius veiksmus.

Galutinis vartotojas turi galimybę bet kada prisijungti prie testuojamosios sistemos versijos ir ją išmėginti. Kiekvieno susitikimo ar vaizdo konferencijos metu vartotojas išreiškia pastebėjimus ir pasiūlymus, įvertina, ar sąsaja atitinka reikalavimus.

Pagal atsiliepimus, sistemos išvaizda patraukli, neperkrauta. Lengva greitai atlikti norimas užklausas, nes funkcijos sugrupuotos ir pateikiamos šoniniame meniu. Kiekvienas meniu elemento pavadinimas aiškus, neklaidinantis ir nedviprasmiškas, todėl greitai nuveda prie norimo pasirinkimo. Atvėrus kurios nors posistemės langą, nėra nereikalingos ar išsibarsčiusios informacijos, visi reikalingi duomenys pateikiami jiems priklausančioje vietoje. Yra ir vaizdiniai pagalbininkai – ikonėlės, kurios labiau įgudusiam vartotojui pagreitina paiešką, leisdamos neskaityti pavadinimo, o verčiau paspausti ant pažįstamo paveikslėlio (4 pav.).



### 2.3.8. Duomenų kontrolė

Šiame skyrelyje pateikiama įvairių duomenų įvedimo bei išvedimo kontrolė, aprašomi atvejai, kurių metu neprieinami kai kurie duomenys, užklaustos, kurios negali būti įvykdytos dėl neteisingo įvedimo (4 lentelė).

Atostogų prašymai
<p>Sukauptus viršvalandžius galima panaudoti tik tuo atveju, jei pasirinktas atostogų tipas yra "Annual" (kasmetinės), nes už viršvalandžius gaunamos atostogos, kaip ir sukauptos kasmetinės atostogos, yra vieno tipo - priklauso už tuo metus dirbtą darbo laiką. Jei darbuotojas sukauptų viršvalandžių neturi, nepateikiama net viršvalandžių pasirinkimo įvestis (apsauga realizuota ir mikroservise).</p> <p>Administratorius gali matyti savo, skirtingų klientų (komandų) bei apskritai visų darbuotojų atostogų prašymus lentelėje ir kalendoriuje. Darbuotojas gali matyti tik savo ir savo komandos atostogų prašymus.</p> <p>Galioja atostogų datos pasirinkimo ribojimai (6 pav.):</p> <ul style="list-style-type: none"><li>• pasirinkta pradžios ar pabaigos data negali būti savaitgalis.</li><li>• pasirinkta pradžios ar pabaigos data negali būti šiandiena arba ankstesnė.</li><li>• pasirinkta pradžios ar pabaigos data negali būti šventė.</li></ul> <div data-bbox="676 936 1120 1476"></div> <p>6 pav. Atostogų datos pasirinkimo ribojimai</p> <p>Nėra ribojama galimybė atostogų prašyti į priekį. Jei nėra sukaupta pakankamai atostogų, darbuotojas vis tiek gali pateikti prašymą. Po prašymo patvirtinimo darbuotojas turės neigiamą atostogų rezervą.</p>
Darbuotojai
<p>Pridėti, redaguoti ar ištrinti darbuotoją gali tik administratorius. Administratoriui neleidžiama ištrinti savęs ar pakeisti savo statuso į „Former“ (buvęs darbuotojas). Bandant ištrinti bet kurią paskyrą, pasirodo pranešimai, prašantys patvirtinimo.</p> <p>Administratorius gali pakeisti darbuotojo slaptažodį, bet tik tuo atveju, jeigu darbuotojo statusas „Current“ (esamas darbuotojas), o ne „Former“ (buvęs darbuotojas).</p>

4 lentelė. Duomenų kontrolė



## 2.4. Testavimo eiga ir rezultatai

Šiame skyriuje pateikiami sistemos testavimo eiga ir rezultatai.

Programos testavimui (konkrečiau – jos mikroservisams), kuriami vienetų testai su „xUnit“ biblioteka. Ši biblioteka yra nemokama, atviro kodo ir skirta būtent C# kalba parašytoms .NET karkaso programoms.

Vienetų testams sukurtas atskiras projektas, testuojami visi mikroservisų metodai, iš viso parašyti 233 testai. Jų paskirtis svyruoja nuo nesudėtingų paieškos pagal identifikatorių ar įrašo šalinimo metodų iki laiškų siuntimo, dokumentų generavimo metodų testavimo.

Gausiai testuoti nuspręsta todėl, kad:

- tai leidžia išsamiai patikrinti kokybę, atrasti daugumą defektų ir išvengti nenumatytų sunkumų programos tolesniame vystyme;
- jei reikia dažnai tikrinti rezultatus, sutaupoma daug laiko, nes naujas pilnos programos perkompiliavimas trunka daug ilgiau nei vieno testo;
- kiekvienas praktiką atliekantis studentas turi galimybę susipažinti su įvairiausiais testavimo subtilumais (dauguma skirtingų mikroservisų reikalauja ir skirtingų testavimo metodų).

Testuojama ne su reliacine sistemos duomenų baze ir realiais joje saugomais duomenimis, o su kompiuterio „In-Memory“ duomenų baze. Testavimo metu visi testiniai duomenys, surašyti „Set-up.cs“ faile, inicializuojami ir patalpinami kompiuterio operatyviojoje (RAM) atmintyje. Inicializavimas vykdomas prieš kiekvieną naują testą, jo metu duomenimis įvairiai manipuluojama, o testui pasibaigus tie patys duomenys atstatomi ir paruošiami sekančiam testui. Tokios funkcijos kaip laiškų siuntimas ar failų kūrimas testavimo metu yra pamėgdžiojamos, pasinaudojant biblioteka „Moq“.

Vartotojo sąsaja testuojama pagal žemiau pateiktus scenarijus (5 lentelė).

1-o testo scenarijus. Prisijungimas prie sistemos ir naujos paskyros užregistravimas
<p>Aprašymas: prie sistemos prisijungia administratorius, naudodamas diegimo metu sukurtą paskyrą. Sėkmingai prisijungus, galima pridėti naujus vartotojus.</p> <p>Pasiruošimas testui:</p> <ul style="list-style-type: none"><li>• pradinė vartotojo rolė – svečias,</li><li>• prisijungimo duomenys jau diegimo metu patalpinti į duomenų bazę,</li><li>• paskyros, su kuria jungiamasi, rolė - administratorius.</li></ul> <p>Testo eiga:</p> <ol style="list-style-type: none"><li>1) Į naršyklę įvesti svetainės adresą ir atverti puslapį. Pasirodys prisijungimui skirtas langas.</li><li>2) Įvesti dokumentacijoje nurodytus el. pašto adresą ir slaptažodį, spausti "Login".</li><li>3) Įsitikinti, kad atsivėrusiame puslapyje prieinamas funkcionalumas, reikalingas administratoriui: puslapio meniu turi sudaryti nuorodos į darbuotojų, klientų, kalendoriaus, profilio, pranešimų šablonų, atostogų ir inventoriaus puslapius.</li><li>4) Iš šoninio meniu pasirinkti "Employees". Atsivers puslapis su darbuotojų sąrašu.</li><li>5) Paspausti mygtuką "Add new employee", kuris yra virš sąrašo lentelės.</li></ol>

6) Į atsivėrusią formą suvesti prašomus duomenis. Nebūtinai visi turi būti rišlūs ir teisingi. Svarbiausia suvesti įsimenamą el. paštą ir slaptažodį. Pasirinkti, kad "Role" būtų "Employee". 7) Paspausti mygtuką "Add employee", bus užregistruotas naujas vartotojas. 8) Lentelė automatiškai atsinaujins. Patikrinti, ar atsirado naujo vartotojo įrašas. 9) Iš šoninio meniu pasirinkti "Profile", "Sign out". Taip bus atsijungta nuo sistemos ir vėl pasirodys prisijungimo langas. 10) Suvesti naujai sukurto vartotojo prisijungimo duomenis. 11) Kadangi užregistruoto vartotojo rolė - "Employee", šoniniame meniu turėtų būti matomi darbuotojui skirti pasirinkimai: kalendorius, atostogos, profilis.
<b>2-o testo scenarijus. Įmonės atostogų teikimo tvarkos dokumentas, jo įkėlimas ir peržiūra</b>
<p>Aprašymas: bet kuris sistemos administratorius turi galėti pakeisti atostogų teikimo tvarkos dokumentą (įkelti naują), bet kuris prisijungęs vartotojas turi galėti šį dokumentą peržiūrėti.</p> <p>Pasiruošimas testui:</p> <ul style="list-style-type: none"> <li>prie sistemos prisijungta su administratoriaus paskyra,</li> <li>paruošti du .pdf failai: vienas iš jų - dabartinis atostogų teikimo tvarkos dokumentas, kitas - bet koks (patartina nedidelės apimties) .pdf failas.</li> </ul> <p>Testo eiga:</p> <ol style="list-style-type: none"> <li>Šoniniame meniu pasirinkti "Profile", "Settings".</li> <li>Atsivers puslapis su dviem langeliais, skiriančiais pranešimų nustatymus nuo bendrų nustatymų. Paspausti bendrų nustatymų ("General settings") langelyje esantį mygtuką "Upload holiday policy".</li> <li>Atsivėrusiame modaliniame lange spausti "Upload". Atsidarys kompiuterio failų naršyklė. Pasirinkti naują .pdf failą.</li> <li>Testas pavykęs, jeigu naują .pdf dokumentą galima peržiūrėti kaip atostogų teikimo tvarkos dokumentą. Patikrinti nuėjus į pradinį "Home" puslapį, paspaudus ant mygtuko "View holiday policy".</li> </ol> <p>Sutvarkymas po testo:</p> <ul style="list-style-type: none"> <li>kartojant 1-3 žingsnius, vietoj testinio dokumento įkelti buvusį dokumentą.</li> </ul>
<b>3-io testo scenarijus. Naujų duomenų bazės įrašų kūrimas</b>
<p>Aprašymas: bet kuris sistemos administratorius turi galėti pridėti naujus klientų, darbuotojų ir inventoriaus įrašus.</p> <p>Pasiruošimas testui:</p> <ul style="list-style-type: none"> <li>prie sistemos prisijungta su administratoriaus paskyra.</li> </ul> <p>Testo eiga:</p> <ol style="list-style-type: none"> <li>Šoniniame meniu pasirinkti "Clients", bus atvertas klientų sąrašo puslapis.</li> <li>Spausti ant mygtuko "Add client", esančio pačiame puslapio viršuje, kairėje pusėje. Atsivers modalinis langas su naujo įrašo sukūrimo forma.</li> <li>Atsivėrusią formą užpildyti reikalingais duomenimis (nebūtinai rišliais ir teisingais).</li> <li>Nepalikus formos, tame pačiame modaliniame lange spausti mygtuką "Add client".</li> <li>Patikrinti, ar automatiškai atsinaujinusioje lentelėje atsirado naujai sukurtas įrašas.</li> <li>1-5 žingsnius pakartoti su "Employees" ir "Inventory" puslapiais.</li> </ol>

4-o testo scenarijus. Esamų duomenų bazės įrašų redagavimas
<p>Aprašymas: bet kuris sistemos administratorius turi galėti redaguoti esamus klientų, darbuotojų ir inventorius įrašus.</p> <p>Pasiruošimas testui:</p> <ul style="list-style-type: none"> <li>• "Clients", "Employees", "Inventory" lentelėse yra bent po vieną įrašą.</li> </ul> <p>Testo eiga:</p> <ol style="list-style-type: none"> <li>1) Šoniniame meniu pasirinkti "Clients", bus atvertas klientų sąrašo puslapis.</li> <li>2) Kiekvieno įrašo eilutėje, "Actions" stulpelyje yra mygtukas "Edit". Paspaudus ant šio mygtuko, atsidarys modalinis langas su esamo įrašo redagavimo forma.</li> <li>3) Kiekvienas įvedimo langas užpildytas su esamais duomenimis. Atlikti bent nežymų duomenų pakeitimą kiekviename lauke.</li> <li>4) Nepalikus formos, tame pačiame modaliniame lange spausti mygtuką "Edit".</li> <li>5) Patikrinti, ar atnaujinto įrašo duomenys sutampa su įvestais redagavimo formoje.</li> <li>6) 1-5 žingsnius pakartoti su "Employees" ir "Inventory" puslapiais.</li> </ol>
5-o testo scenarijus. Gimtadienio matomumo nustatymas
<p>Aprašymas: bet kuris registruotas vartotojas turi teisę nustatyti, ar kitiems sistemos naudotojams bus pranešama apie jo gimtadienį. Privatūs gimtadieniai kalendoriuje turi būti matomi tik administratoriui, su ženkliniu, kad gimtadienis privatus.</p> <p>Pasiruošimas testui:</p> <ul style="list-style-type: none"> <li>• darbuotojų lentelėje yra bent du įrašai,</li> <li>• vieno iš darbuotojų gimtadienis švenčiamas einamą mėnesį.</li> </ul> <p>Testo eiga:</p> <ol style="list-style-type: none"> <li>1) Prisijungti su administratoriaus paskyra.</li> <li>2) Atverti kalendorių ("Calendar", pasirinkimas šoniniame meniu).</li> <li>3) Patikrinti, ar kalendoriuje matomas einamą mėnesį švenčiamas darbuotojo gimtadienis. Taip pat, įvykis neturi būti pažymėtas kaip privatus.</li> <li>4) 2-3 žingsnius pakartoti prisijungus su vieno iš darbuotojų paskyra.</li> <li>5) Prisijungti prie gimtadienį švenčiančio darbuotojo paskyros.</li> <li>6) Šoniniame meniu pasirinkti "Profile", "Settings".</li> <li>7) Atsivers puslapis su dviem langeliais, skiriančiais pranešimų nustatymus nuo bendrų nustatymų. Paspausti pranešimų nustatymų ("Notification settings") langelyje išjungti "Broadcast birthday notifications".</li> <li>8) Prisijungti su administratoriaus paskyra, atverti kalendorių. Dabar gimtadienio įvykis turi būti pažymėtas kaip privatus.</li> <li>9) Prisijungti su bet kurio kito darbuotojo paskyra, atverti kalendorių. Darbuotojo rolės vartotojui neturi būti matomas privatus gimtadienis.</li> </ol>
6-o testo scenarijus. Nuosavo slaptažodžio keitimas
<p>Aprašymas: bet kuris registruotas vartotojas turi teisę pasikeisti slaptažodį, su kuriuo prisijungia prie paskyros. Senasis slaptažodis turi nesuteikti galimybės prisijungti.</p> <p>Pasiruošimas testui:</p> <ul style="list-style-type: none"> <li>• prie sistemos prisijungta su bet kurio registruoto vartotojo paskyra</li> </ul>

Testo eiga:

- 1) Šoniniame meniu pasirinkti "Profile", "Settings".
- 2) Atsivers puslapis su dviem langeliais, skiriančiais pranešimų nustatymus nuo bendrų nustatymų. Paspausti bendrų nustatymų ("General settings") langelyje esantį mygtuką "Change password".
- 3) Įvesti naują slaptažodį ir nepalikus formos, tame pačiame modaliniame lange spausti mygtuką "Change password".
- 4) Atsijungti nuo sistemos.
- 5) Prisijungimo lange įvesti buvusį slaptažodį ir spausti "Login". Turi būti matomas pranešimas apie neteisingai suvestus prisijungimo duomenis.
- 6) Prisijungimo lange įvesti naują slaptažodį ir spausti "Login". Prisijungimas prie paskyros turi būti sėkmingas.

#### **7-o testo scenarijus. Atostogų prašymo pateikimas ir patvirtinimo procesas**

Aprašymas: kiekvienas registruotas vartotojas turi teisę pateikti atostogų prašymą, į kiekvieno administratoriaus el. pašto dėžutę turi būti išsiunčiami pranešimai apie prašymo patvirtinimą. Laukiant, atmetus, patvirtinus, ar pačiam darbuotojui apleidus prašymą, turi būti matomas statuso pasikeitimas.

Pasiruošimas testui:

- visų registruotų administratorių el. pašto adresai yra teisingi, turima prieiga prie kiekvienos registruotos pašto dėžutės,
- bent vieno iš darbuotojų el. pašto adresas turi būti teisingas - būtent šio darbuotojo paskyrą naudoti testo vykdymui.

Testo eiga:

- 1) Prisijungti su darbuotojo paskyra.
- 2) Šoniniame meniu pasirinkti "Holidays", bus atvertas atostogų prašymų sąrašas.
- 3) Spausti puslapio viršuje, kairėje pusėje esantį mygtuką "Request holidays".
- 4) Užpildyti ir pateikti naują prašymą.
- 5) Pagal 3-4 žingsnius pateikti dar du prašymus. Kiekvieno iš jų statusas turi būti "Pending" (laukiantis patvirtinimo).
- 6) Vieną iš prašymų apleisti (spausti įrašui priklausanti mygtuką "Abandon", esanti "Action" stulpelyje). Statusas turi pasikeisti į "Abandoned".
- 7) Atverti vieną iš el. pašto dėžučių, susietų su administratoriaus paskyra. Turi būti atėję trys laiškai apie prašomą patvirtinimą (po vieną laišką kiekvienam prašymui).
- 8) Paspaudus ant atsiųstos nuorodos, atmesti prašymą. Kitą prašymą patvirtinti. Atvėrus vieną iš nuorodų turi būti matoma, kad darbuotojas apleido prašymą ir patvirtinimas nebereikalingas.
- 9) Prisijungus per atostogų prašymus sukūrusio darbuotojo arba bet kurio administratoriaus paskyrą ir atvėrti atostogų lentelę. Turi būti matoma, kad vienas iš prašymų buvo atmestas, kitas - patvirtintas.
- 10) Šoniniame meniu pasirinkus kalendorių, atvertame lange turi būti matomas tik patvirtintas atostogų įrašas.
- 11) Atidarius pašto dėžutę, priklausančią darbuotojui, turi būti atsiųstas patvirtintos atostogos prašymo dokumentas, automatiškai užpildytas ir paruošti spausdinimui.

#### **8-o testo scenarijus. Įrašų paieška ir šalinimas**

Aprašymas: turi būti galima ieškoti lentelės įrašo pagal įvestą paieškos tekstą bei ištrinti norimą lentelės įrašą.

Pasiruošimas testui:

- "Clients", "Employees", "Inventory" ir "Holidays" lentelėse yra bent po vieną įrašą,
- prie sistemos prisijungta su bet administratoriaus paskyra.

Testo eiga:

- 1) Šoniniame meniu pasirinkti "Clients", bus atvertas klientų sąrašo puslapis.
- 2) Virš lentelės su įrašais yra įvedimo laukas "Search". Įvesti paieškos tekstą, kuris matomas bet kuriame įrašė. Lentelėje turi likti tik įrašas, kuriame yra norimas paieškos tekstas.
- 3) Kiekvieno įrašo eilutėje, "Actions" stulpelyje yra mygtukas "Edit". Paspaudus ant jo, atsidarys modalinis langas su redagavimo forma.
- 4) Formos apačioje yra mygtukas "Delete". Paspaudus šį mygtuką, iš atsinaujinusios lentelės turi būti pašalintas susijęs įrašas.
- 5) Pakartoti 1-4 žingsnius su "Employees", "Inventory" ir "Holidays" puslapiais.

#### **9-o testo scenarijus. Viršvalandžiai ir atostogų dienų nuskaitymas**

Aprašymas: kiekvienas registruotas vartotojas, turintis sukauptų viršvalandžių, gali juos panaudoti prašydamas atostogų. Atostogos su viršvalandžiais žymimos atskira spalva kalendoriuje, apie viršvalandžių panaudojimą informuojama papildomais sakiniais prašymo laiške, prašymo formoje.

Pasiruošimas testui:

- visų registruotų administratorių el. pašto adresai yra teisingi, turima prieiga prie kiekvienos registruotos pašto dėžutės,
- bent vieno iš darbuotojų el. pašto adresas turi būti teisingas - būtent šio darbuotojo paskyrą naudoti testo vykdymui,
- darbuotojas turi bent 20 sukauptų viršvalandžių (galima redaguoti tiesiogiai duomenų bazėje arba prisijungus su administratoriaus paskyra).

Testo eiga:

- 1) Prisijungti su darbuotojo paskyra.
- 2) Pradiniame puslapyje matoma informacija apie sukauptas atostogų dienas bei viršvalandžius, kuriuos galima panaudoti atostogoms. Įsiminti esamus skaičius.
- 3) Šoniniame meniu pasirinkti "Holidays", bus atvertas atostogų prašymų sąrašas.
- 4) Spausti puslapio viršuje, kairėje pusėje esantį mygtuką "Request holidays".
- 5) Pasirinkti atostogų tipą "Annual", prašyti iš viso keturių atostogų dienų. Pasirinkti, kad dvi atostogų dienos - už sukauptus viršvalandžius.
- 6) Patvirtinti prašymą per gautą laišką, išsiųstą visiems administratoriams. Laiške turi būti nurodyta, kad norima panaudoti dvi viršvalandinės dienos.
- 7) Atvėrus pradinį puslapį ir pažiūrėjus į sukauptų atostogų dienų informaciją, turi būti matoma, kad nuskaičiuotos dvi bendros atostogų dienos ir dvi viršvalandinės.
- 8) Atvėrus kalendorių, atostogų įrašas nuspalvintas spalva, žyminčia "Annual paid, with overtime" (apmokamos, su viršvalandžiais) atostogų tipą.
- 9) Į darbuotojo el. pašto dėžutę atsiųsti spausdinimui paruošti dokumentai taip pat turi informuoti apie panaudotus viršvalandžius.

*5 lentelė. Testų scenarijai*

## 2.5. Dokumentacija

Šiame skyriuje pateikiama sukurto produkto galutinio vartotojo ir programuotojo dokumentacija.

### Dokumentacija programuotojui Vartotojo sąsajos paruošimas

Pirmą kartą atidarius *Visual Studio Code*, reikia:

- Atverti terminalą (įrankių juostoje, „New Terminal“).
- Įvesti **npm install -g @angular/cli** ir spausti Enter.
- Rasti aplanką, kuriame yra Angular projekto failas (**ClientApp**, jame turi būti **src** ir **e2e** aplankai) – tai projekto direktorijos aplankas. Visur, kur toliau minimas 'ClientApp', reikia įvesti pilną aplanko kelią.
- Terminale įvesti:

```
cd 'ClientApp'
npm install
ng serve
```
- Atverti savo interneto naršyklę.
- Adreso laukelyje įvesti RootUrl (numatytoji: <https://localhost:4200>), spausti Enter. Jeigu RootUrl pasikeitęs, naujojo ieškoti **XplicityApplication\Xplicity Holidays** (mikroservisų projekte), **appsettings.json** faile, **AppSettings** skiltyje. Naršyklei baigus krauti puslapį, turėtų matytis prisijungimo langas, tačiau be galimybės su juo sąveikauti.

### Mikroservisų paruošimas

- Atidaryti **XplicityApplication\Xplicity Holidays** aplanke esantį projekto failą **Xplicity Holidays.sln**. Projektas bus atvertas *Visual Studio 2019* programavimo aplinkoje.
- Atverti terminalą (įrankių juostoje, View > Other Windows > Package Manager Console).
- Įvesti **Update-Database** ir spausti Enter. Pagal migracijų failus duomenų bazėje bus sukurti reikiami modeliai.
- Spausti CTRL+F5, projektas bus sukompiliuotas.

Kaip pasiekti duomenų bazės lenteles rankiniam redagavimui?

Atverti *Microsoft SQL Server Management Studio 18* ir prisijungti su šiais nustatymais:

- Server type: Database Engine
- Server name: localhost\sqlexpress (patikrinti **appsettings.json** failą dėl pakeitimų, turėtų būti **localhost\SQLEXPRESS**)
- Authentication: Windows Authentication

### Dokumentacija vartotojui

- Atverti savo interneto naršyklę, į adreso laukelį įvesti įdiegtos ir naudojimui paruoštos svetainės adresą: <https://xplicityapp.azurewebsites.net/>
- Į atsivėrusį prisijungimo langą įvesti duotą el. pašto adresą ir slaptažodį.
- Prisijungus atsivers namų puslapis su šoniniu meniu, iš kurio galima nueiti į bet kurį darbuotojui ar administratoriui skirtą puslapį.
- Dėl išsamaus aprašymo, kaip naudotis kiekviena sistemos funkcija, skaityti testavimo scenarijus skyrelyje „[Testavimo eiga ir rezultatai](#)“.

## Išvados

Praktikos metu kuriant atostogų apskaitos informacinę sistemą, buvo pasiekti visi pradžioje užsibrėžti tikslai. Praktikos rezultatai tokie, kokie ir buvo numatyti: viršvalandžių posistemė leidžia registruoti darbuotojo sukauptus viršvalandžius, rodo informaciją apie jų likutį, leidžia viršvalandžius panaudoti prašant apmokamų atostogų dienų. Visa informacija apie panaudotus viršvalandžius pateikiama atostogų prašymų lentelėje, el. pranešimuose, prašymo ir patvirtinimo dokumentuose.

Projektas neturėjo jokių projektavimo modelių, vartotojo sąsajos eskizų ar testavimo scenarijų, todėl viskas buvo parašyta ir nubraižyta praktiką atlikusios studentės, kai projektas ir jam keliami reikalavimai buvo galutinai perprasti ir pažįstami. Pavyko rasti produkto konkurentus, jų įkainius ir surašyti funkcijas, kurias gali pasiūlyti tik UAB „Xplicity“ vystoma sistema.

Labiausiai džiugu dėl šių pasiektų asmeninių rezultatų:

- Neapsiribota vien tuo, kas buvo pirmiausiai išmokta ir nesudėtingai atliekama. Stengtasi rinktis įvairias užduotis ir dirbti su visa sistemai reikalinga programine įranga.
- Siekta, kad žinios apie projektą išliktų aktualios ir gausios, su kuo mažiau spragų. Tai reikalinga, kad būtų gebama pakomentuoti iškilus neaiškumams ar ieškant tolesnių sprendimų, netyčia nebūtų suklaidinti kiti komandos nariai.
- Pagilintos testavimo žinios rašant vienetų testus įvairioms funkcijoms. Stengiantis padengti kuo daugiau kodo eilučių, pavyko išsiaiškinti apie skirtingų testų subtilybes.

Neigiamos savybės, kurias teko pastebėti savo darbe:

- Esant galimybei peržiūrėti kitų komandos narių darbus, tai ne visada pavykdavo atlikti. Tai svarbu, nes kuo daugiau komandos narių peržiūti vieni kitų darbus, tuo lengviau išvengti tolesnio darbo apsunkinimo ir klaidų.
- Kartais trūko aktyvumo išreiškiant savo sprendimus ir pasiūlymus.

Nepaisant to, kad nevisada pavykdavo pasiūlyti geriausią sprendimą bendroms projekto užduotims, visos suplanuotos užduotys buvo atliktos laiku ir pagal visus reikalavimus. Pagal praktikos vadovo atsiliepimus, indėlis į vartotojo sąsajos kūrimą smarkiai pagerino vartotojų pasitenkinimą sistema.

Visos užduotys, kurias teko atlikti, stipriai pagilino Angular ir .NET Core žinias. Žinant, kad šios technologijos yra dažnai naudojamos IT įmonėse, galima teigti, kad atlikta praktika suteikė naudingų ir perspektyvių programavimo žinių.

## Literatūra

1. SimpleApprove (2020 m.). Nuskaityta 2020 m. kovo 13 d. iš „Patogus personalo dokumentų valdymas – Simpleapprove.lt“: <http://www.simpleapprove.lt/>
2. E-atostogos, HCM.LT (2017 m.). Nuskaityta 2020 m. kovo 13 d. iš „Elektroninės atostogos – atostogų planavimo ir valdymo sistema online“: <https://e-atostogos.lt/>
3. Calamari (2020 m.). Nuskaityta 2020 m. kovo 13 d. iš „Leave management and attendance tracking system – Calamari“: <https://calamari.io/>
4. CakeHR (2020 m.). Nuskaityta 2020 m. kovo 14 d. iš „CakeHR | HR Management Software“: <https://cake.hr/>
5. Microsoft Docs (2020 m. vasario 12 d.). Nuskaityta 2020 m. kovo 16 d. iš „Choose between ASP.NET 4.x and ASP.NET Core | Microsoft Docs“: <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/choose-aspnet-framework?view=aspnetcore-3.1>
6. Yalantis, Liliia Harkushko (be datos). Nuskaityta 2020 m. kovo 16 d. iš „When and Why Angular Is a Good Technical Solution for Your Project“: <https://yalantis.com/blog/when-to-use-angular/>
7. Microsoft (2020 m.). Nuskaityta 2020 m. kovo 16 d. iš „SQL Server Downloads“: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads-free-trial>
8. Microsoft Docs (2018 m. gruodžio 20 d.). Nuskaityta 2020 m. kovo 16 d. iš „Agile process work item types and workflow“: <https://docs.microsoft.com/en-us/azure/devops/boards/work-items/guidance/agile-process-workflow?view=azure-devops>
9. Darius Šilingas, No Magic Europe (2015 m.). Nuskaityta 2020 m. kovo 16 d. iš „UML modeliavimo kalbos panaudojimas, projektuojant programinę įrangą“: <http://www.nomagic.lt/straipsniai/projektavimas.html>