

**Kauno technologijos universitetas**

Informatikos fakultetas

## **Konkursų organizavimo sistemos statybų įmonėms plėtojimas**

Baigiamosios praktikos ataskaita

---

**Matas Žilinskas**

Studentas

2021-04-23

**Gintautas Kučinskas**

Vadovas

**Doc. Giedrius Ziberkas**

Dėstytojas

---

## Turinys

<b>Įvadas.....</b>	<b>4</b>
<b>1. Analizės dalis .....</b>	<b>5</b>
1.1. Informacija apie įmonę.....	5
1.1.1. Įmonės veiklos sritis, specializacija .....	5
1.1.2. Įmonės vizija .....	5
1.1.3. Įmonės misija .....	5
1.1.4. Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją .....	5
1.2. Informacija apie įmonėje naudojamą IT .....	5
1.3. Įmonės veiklos procesai. ....	6
1.3.1. Projekto inicijavimas.....	6
1.3.2. Projekto koncepcijos analizė.....	6
1.3.3. Kainos ir investicijų analizė.....	6
1.3.4. Projektų vykdymas, pristatymas ir dokumentacija. ....	6
1.3.5. Komandinis darbas.....	6
1.3.6. Inovacijos. ....	6
1.3.7. Produktų vystymo procesai. ....	7
<b>2. Praktikos dalis .....</b>	<b>8</b>
2.1. Techninė užduotis .....	8
2.1.1. Sistemos apibrėžimas .....	8
2.1.2. Bendras veiklos tikslas .....	8
2.1.3. Sistemos pagrįstumas .....	8
2.1.4. Konkurencija rinkoje.....	8
2.1.5. Prototipai ir pagalbinė informacija.....	8
2.1.6. Siūlomas sprendimas.....	8
2.1.7. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti .....	8
2.2. Reikalavimų specifikacija .....	9
2.2.1. Komeracinė specifikacija.....	9
2.2.2. Sistemos funkcijos .....	10
2.2.3. Apribojimai .....	13
2.2.4. Duomenų ir rezultatų specifikacija .....	13
2.2.5. Vartotojo sąsajos specifikacija .....	13
2.2.6. Realizacijai keliami reikalavimai. ....	14
2.2.7. Techninė specifikacija.....	14
2.3. Projektavimo eiga ir sistemos projektas.....	15
2.3.1. Projektavimo valdymas ir eiga.....	15
2.3.2. Projektavimo technologija .....	15
2.3.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos .....	15
2.3.4. Informacijos srautai.....	15
2.3.5. Sistemos sudėtis .....	16
2.3.6. Komponentai .....	16
2.3.7. Sąsajos įvertinimas pagal vartotojo patyrimą, profesinę terminologiją .....	16
2.3.8. Duomenų kontrolė.....	17

2.4. Testavimo eiga ir rezultatai .....	17
2.5. Dokumentacija .....	17
<b>Išvados .....</b>	<b>18</b>
<b>Literatūra .....</b>	<b>19</b>

## **Ivadas**

Baigiamoji praktika buvo atliekama įmonėje UAB „Xplicity“. Tai Olandijos kilmės įmonė, kuri turi padalinius Kaune bei Amsterdame. „Xplicity“ kuria bei palaiko įvairią programinę įrangą, rengia praktikas universitetams bei akademijas naujiems programuotojams.

Praktikos dalykinė sritis – reali, klientų užsakyta internetinė svetainė, skirta valdyti statybų konkursams.

Praktikos tikslas – plėsti bei tobulinti informacinę sistemą, kurioje naudotojai turi turėti galimybę dalyvauti statybų konkursuose.

Numatomi praktikos rezultatai sudaryti tokie:

- Sistemos naudotojas gali manipuluoti dokumentais.
- Sistemos naudotojas gali priimti svetimą konkursą.
- Sistemos naudotojas gali paskelbti kainos pasiūlymą svetimam konkursui.
- Sistemos naudotojas gali keisti puslapiuose naudojamą teksto kalbą.
- Sistemos naudotojas gali pakviesti svetimą įmonę prisijungti prie sistemos.

Šioje praktikos ataskaitoje yra pateikiama UAB „Xplicity“ ir projekto analizė.

## **1. Analizės dalis**

### **1.1. Informacija apie įmonę**

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

UAB „Xplicity“ dirba informacinių technologijų srityje, specializuojasi į internetinių svetainių kūrimą bei vystymą.

#### **1.1.2. Įmonės vizija**

Įmonės vizija – sumažinti tarpą tarp verslo ir technologijų.

#### **1.1.3. Įmonės misija**

Įmonės misija – suteikti kokybiškus programavimo sprendimus savo klientams bei demonstruoti skaidrumą, atsidavimą bei meistriškumą tiek įmonės viduje, tiek išorėje.

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

Įmonė kreipia ypatingą dėmesį į savo darbuotojų komandą:

Visų pirma, „Xplicity“ deda rimtas pastangas į naujų kompetentingų, inovatyvių bei siekiančių tobulėti specialistų paiešką: organizuoja tiek akademiją, tiek darbo praktiką, bendradarbiauja su aukštojo išsilavinimo įstaigomis (tiek Kauno Technologijos Universitetu, tiek Kauno Kolegija). Taipogi, įmonė naudoja jauniems žmonėms aktualiais informacijos šaltiniais – pvz. kelia naujienas į socialinį tinklą „Facebook“.

Visų antra, įmonė skatina esamus darbuotojus tobulėti: organizuoja papildomus apmokymus į juos dominančias technologijų sritis, kviečia atitinkamų sričių specialistus praveisti paskaitų, apdovanoja darbuotojus už rodomą iniciatyvą.

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

Programavimo kalbos:

1. C#
2. PHP
3. Java
4. TypeScript

Naudotojo sąsajos programavimo karkasai: React, Angular.

Versijavimo kontrolės sistema: GIT.

Pagrindinis agile proceso valdymo įrankis: Azure DevOps

### **1.3. Įmonės veiklos procesai.**

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

Projekto inicijavimo etapai:

1. Klientas išvardina norimo įgyvendinti projekto reikalavimus.
2. Įmonė įvertina išvardintus reikalavimus bei pateikia kelias alternatyvas, kaip galima įgyvendinti užsibrėžtą projektą.
3. Klientas išsirenka projekto įgyvendinimo metodą.
4. Įmonė pasiūlo potencialių programuotojų sąrašą projekto įgyvendinimui.
5. Klientas išsirenka darbui tinkamiausius darbuotojus.

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

Siekdami suvokti projekto koncepciją, įmonės atstovai dalyvauja pokalbyje su klientu, per kurį išsiaiškinami aktualūs neaiškumai tiek įmonei, tiek klientui (pvz. funkciniai reikalavimai, pristatymo laikas ir t.t.).

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

Klientas gali matyti jam siūlomų programuotojų valandinius įkainius, kurie buvo nustatyti pagal kompetencijos lygį, bei pats nusprendžia, kiek verta investuoti į darbuotojų kvalifikaciją.

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

Projektai yra vykdomi remiantis Agile [1] principais, dažniausiai pagal Scrum metodologiją. Kiekvieno sprinto pabaigoje klientams yra pristatoma, kas buvo padaryta sprinto metu – taip klientai turi laisvę pakeisti projekto kryptį, jeigu supranta padarę klaidą, kol ji daug nekainavo.

Taip pat projektuose yra plečiamas README.md failas, kuriame aprašytos visos instrukcijos, skirtos paruošti naujo programuotojo darbo aplinką.

#### **1.3.5. Komandinis darbas.**

Projekto komandos vadovaujasi Agile principais (Scrum metodologija). Kiekviena komanda turi savo projekto vadovą (angl. Scrum master), kuris išsprendžia iškilusius nesklandumus bei rūpinasi, kad darbai būtų įvykdyti laiku. Komandos kartu su klientu plečia darbų lentą (angl. Backlog), kurioje yra užduočių aprašymai, bei užduotys yra rikiuojamos pagal svarbą. Sprinto trukmė – dvi savaitės. Jo pradžioje komanda apsitaria, ką ir kaip reikia padaryti. Sprintui pasibaigus, komanda pademonstruoja klientui, ką padarė sprinto metu.

#### **1.3.6. Inovacijos.**

Įmonė aktyviai mėgina plėtotis inovacijų srityje. 2020 m. „Xplicity“ dalyvavo „Hackathon“ renginyje, kuriame skyrė ypatingą dėmesį programos, kuri padeda sumažinti išmetamą maistą bei gėrimus festivaliuose, vystymui.

### 1.3.7. Produktų vystymo procesai.

Produktai yra vystomi pagal Scrum metodologiją:

- Darbas yra suskirstytas į dviejų savaitių iteracijas, kurios yra vadinamos sprintais.
- Sprinto pradžioje klientas pristato funkcinius reikalavimus komandai.
- Po funkcinių reikalavimų pristatymo, komanda apsitaria, kaip juos įvykdyti iš techninės pusės.
- Kiekvieną dieną prieš pradėdami darbą, komandos nariai praneša, ką nuveikė praeitą dieną, ką planuoja nuveikti einamą dieną bei kokių nesklandumų kilo. Jei sunkumų visgi iškilo, komanda bendromis jėgomis juos išsprendžia.
- Kai komandos narys pabaigia darbą (angl. Feature), paviešina savo programinį kodą ir laukia kitų komandos narių kodo peržiūros – gavus jų patvirtinimą, kodas yra suliejamas su pagrindine versijavimo kontrolės šaka.
- Sprinto pabaigoje klientams yra pristatoma, ką komanda padarė sprinto metu, bei visi sudalyvauja retrospektyvoje, kurioje apsitariama kas sprinto metu buvo gerai, kas galėjo būti geriau bei kokių veiksmų reikia imtis, siekiant pagerinti sekantį sprintą.

Tokia eiga iteracijos yra kartojamos vėl iš naujo – taip pamažu produktas yra išvystomas į paruoštą naudoti iki galutinio pristatymo datos.

## **2. Praktikos dalis**

### **2.1. Techninė užduotis**

#### **2.1.1. Sistemos apibrėžimas**

Praktikos metu yra plėtojama konkursų organizavimo statybų įmonėms sistema. Vienos įmonės yra konkursų skelbėjai: gali sukurti projektą, to projekto viduje galima kurti konkursus. Kitos įmonės yra konkursų dalyviai: gali dalyvauti paskelbtuose konkursuose.

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

Sukūrus ir įdiegus sistemą tikimasi, kad ja naudosis įvairios statybų įmonės, kurios klientui generuos pajamas.

#### **2.1.3. Sistemos pagrįstumas**

Kuriamos sistemos reikia todėl, kad statybų įmonės turėtų tarpininką tarp projekto organizatoriaus ir įgyvendintojo.

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

[Aufait](#) [1] – konkursų valdymo sistema, kuri skirta visiems. Mūsų sistema yra konkrečiai skirta statybų sektoriui.

[Odo](#) [2] – konkursų valdymo sistema, kuri yra skirta statybų įmonėms, tačiau visiškai nepritaikyta Lietuvos rinkai.

[Statybų konkursai](#) [3] – konkursų valdymo sistema, kuri skirta statybų įmonėms ir pritaikyta Lietuvos rinkai, tačiau didžiulo projekto (pvz. namo statybos) neina išskaidyti į daug smulkių konkursų (pvz. pamatų įrengimas, langų įmontavimas ir t.t.).

#### **2.1.5. Prototipai ir pagalbinių informacija**

Ši sistema turi seną versiją, įgyvendintą 2015 metais. Naudojama technologija: PHP programavimo kalba serveriui, jQuery biblioteka serverio klientui. Deja, dėl šio sprendimo nebuvo galima įgyvendinti tam tikrų sistemos funkcionalumų. Pagrindinės problemos – nepakankamai moderni technologija naudojama serverio klientui bei yra kelios saugumo spragos serveryje.

#### **2.1.6. Siūlomas sprendimas**

Siūlomas sprendimas: visiškai perdaryti sistemą su kitomis technologijomis: serveriui programuoti bus naudojama C# kalba su ASP.NET karkasu, o serverio klientui programuoti bus naudojama TypeScript kalba su Angular karkasu. Šių technologijų pagalba bus išspręstos minėtos problemos: serverio klientas turės pritraukiančią vartotojo sąsają, o serveris bus stipriai apsaugotas nuo įsilaužimų.

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

Sistemos apimtis: 10 modulių.

Žmogiškieji ištekliai, reikalingi sistemai sukurti: 1 vyresnysis programuotojas ir 1-2 jaunesnieji programuotojai.



## **2.2. Reikalavimų specifikacija**

### **2.2.1. Komerčinė specifikacija**

Projekto užsakovas yra konfidenciali informacija, kurios neturiu teisės viešinti.

Projekto vykdytojai: vienas vyresnysis (angl. Senior) programuotojas ir du jaunesnieji (angl. Junior) programuotojai.

Produkto vartotojai: statybų įmonės.

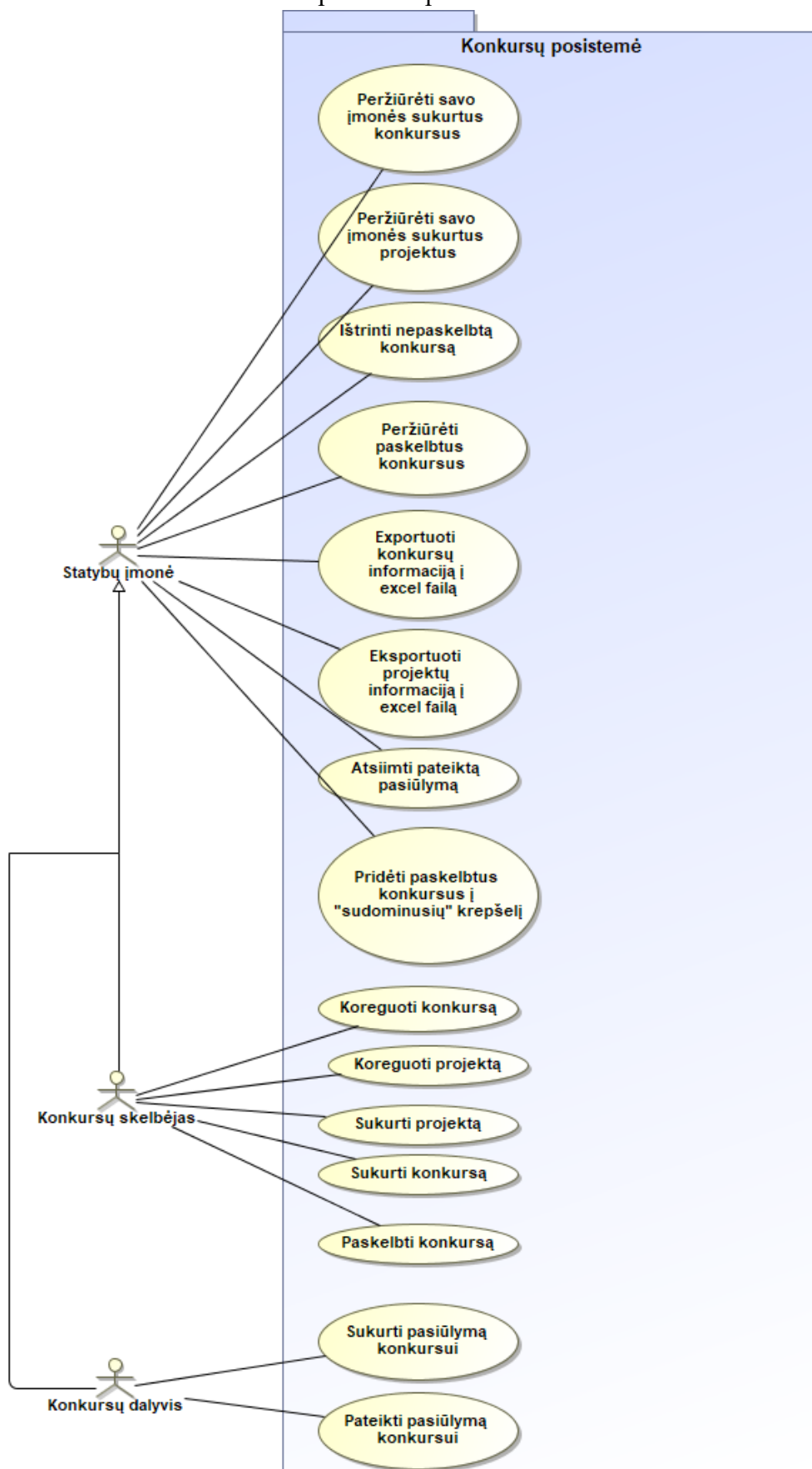
Projekto realizacijos laiko apribojimai: Buvo numatyta projektą vystyti pagal „Agile“ principus, t.y. lanksčiai. Projekto eigoje atsiranda papildomos informacijos, pagal kurią sprendžiama, kaip toliau elgtis – kurią vietą reikia praplėsti, kuri nebuvo numatyta pradžioje. Dėl to ir laiko apribojimai yra lankstūs. Projekto pradžioje numatytas realizacijos laikas buvo 8 mėnesiai, tačiau klientai jį pratęsė.

Projekto realizacijos kainos apribojimai: Konfidencialu.

Pagrindinės priemonės: Serveriui programuoti naudojama C# kalba su ASP.NET karkasu, serverio klientui naudojama TypeScript kalba su Angular karkasu.

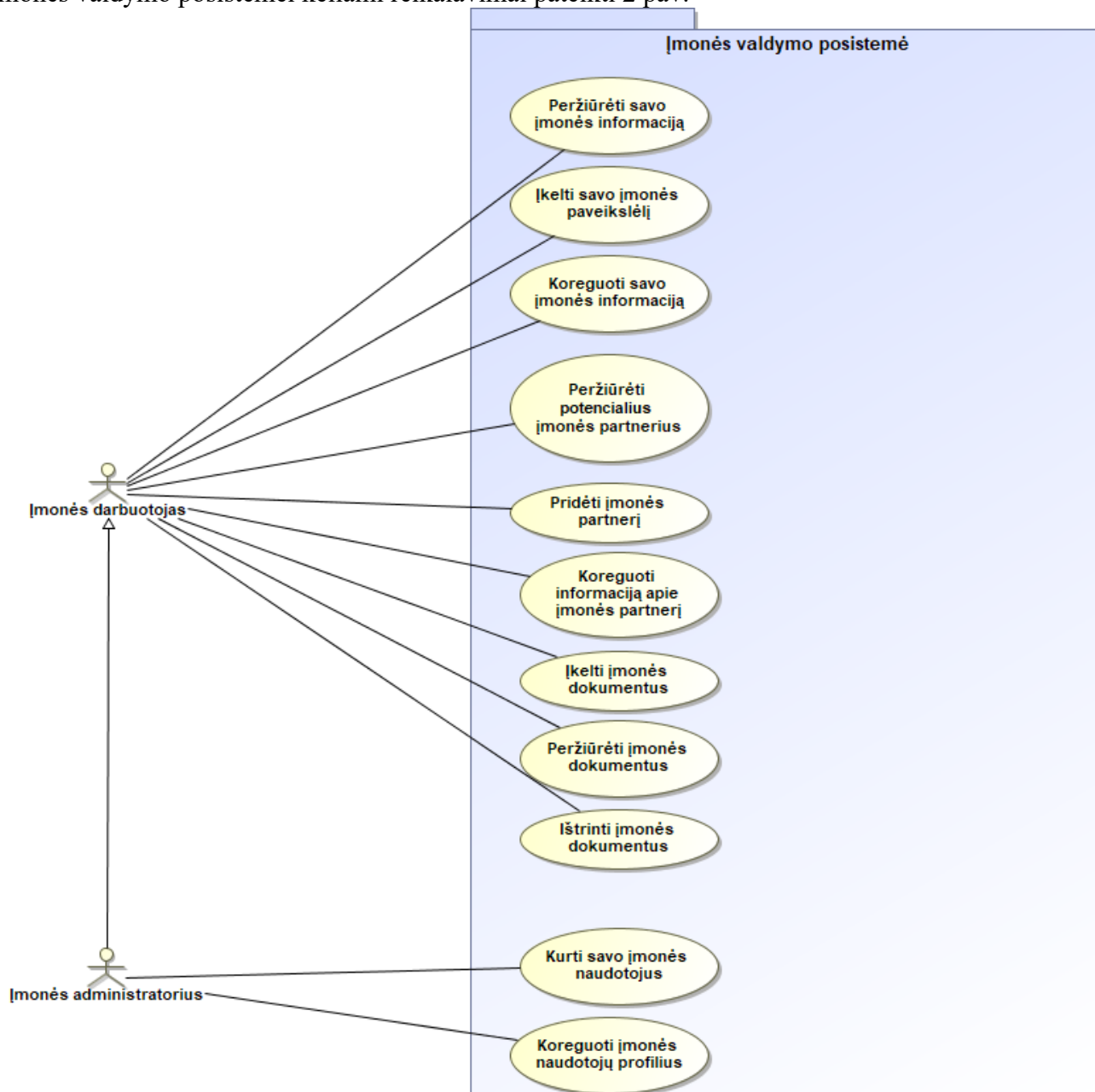
## 2.2.2. Sistemos funkcijos

Konkursų posistemei keliami reikalavimai pateikti 1 pav.



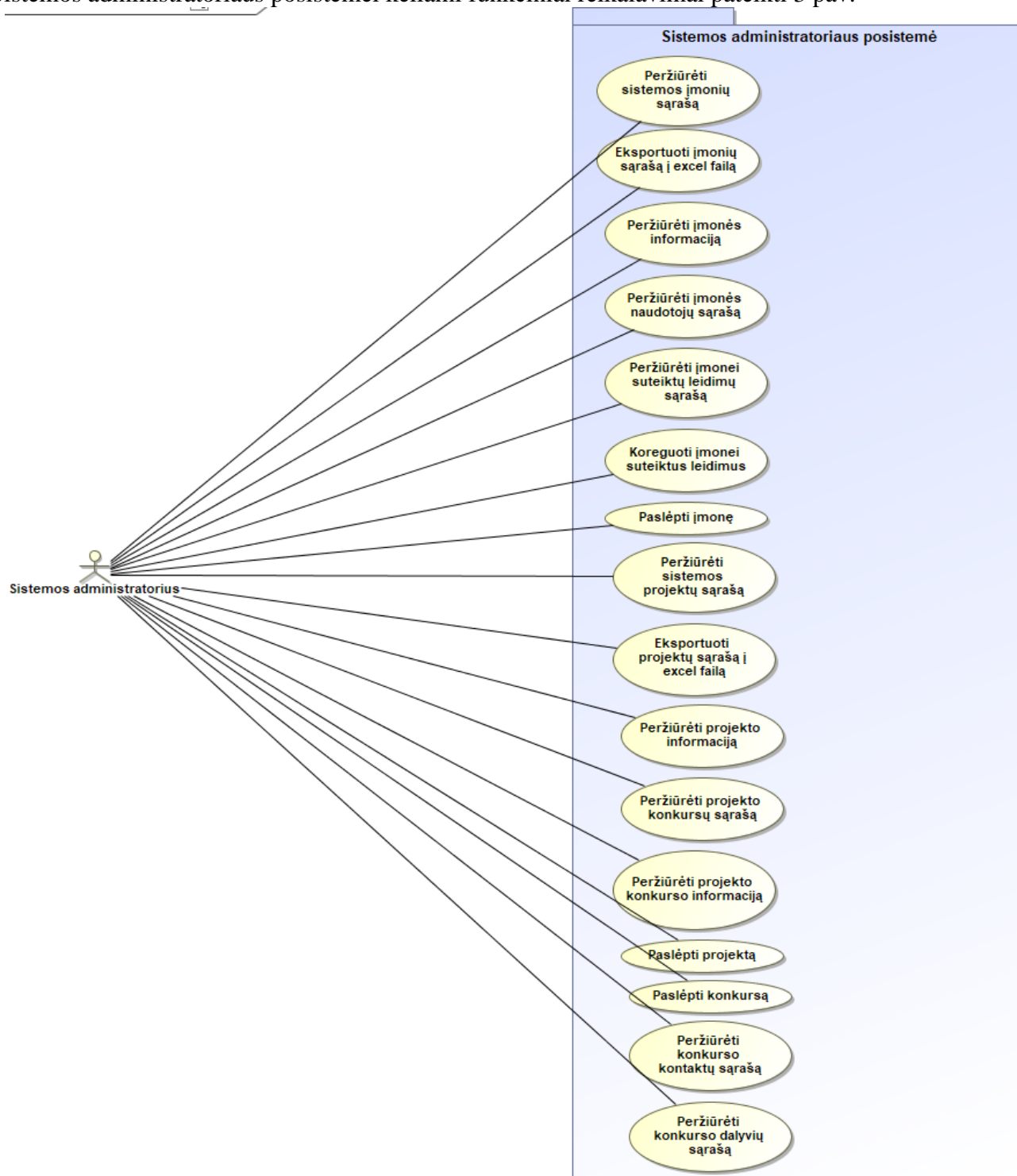
1 pav. Konkursų posistemės funkciniai reikalavimai

Įmonės valdymo posistemei keliami reikalavimai pateikti 2 pav.



**2 pav.** Įmonės valdymo posistemės funkciniai reikalavimai

Sistemos administratoriaus posistemei keliami funkciniai reikalavimai pateikti 3 pav.



**3 pav.** Sistemos administratoriaus posistemė.

### 2.2.3. Apribojimai

Bendradarbiaujančios sistemos: pradinė įmonės informacija turi būti imama iš [rekvizitų](#) vidinės sąsajos.

Darbo vieta: dirbama nuotoliniu būdu.

Darbo laikas: dirbama laisvu grafiku.

Sistemos kūrimo terminai: Konkrečių terminų nėra – projektas nuolat vystomas.

Sistemos kūrimo biudžetas: Konfidencialu.

### 2.2.4. Duomenų ir rezultatų specifikacija

**PASTABA:** Duomenų modeliai buvo supaprastinti su tikslu apsaugoti kliento pranašumą rinkoje.

<b>Projekto modelis</b>	
Pateikimas (GET)	Sukūrimas (POST)
Id: 1 Projekto numeris: 1 Konkursų kiekis: 5 Pavadinimas: Projektas1 Adresas: Kauno g. 1 Aprašymas: Nors koks aprašymas	Pavadinimas: Projektas1 Adresas: Kauno g. 1 Aprašymas: Nors koks aprašymas
<b>Konkurso modelis</b>	
Pateikimas (GET)	Sukūrimas (POST)
Id: 1 Konkurso numberis: 1 Sukūrimo data: 2021-03-04 Pavadinimas: Konkursas1 Konkurso darbai: Darbas1, darbas2, ... Pabaigos data: 2021-04-15	Pavadinimas: Konkursas1 Konkurso darbai: Darbas1, darbas2, ... Pabaigos data: 2021-04-15
<b>Pasiūlymo modelis</b>	
Pateikimas (GET)	Sukūrimas (POST)
Id: 1 Konkurso darbai: Darbas1, darbas2 Užrašai: Bet kokie užrašai Konkurso aprašymas: Nors koks aprašymas Pateiktos kainos darbams: Darbas1: 100, Darbas2: 500, ...	Konkurso darbai: Darbas1, darbas2 Užrašai: Bet kokie užrašai Konkurso aprašymas: Nors koks aprašymas Pateiktos kainos darbams: Darbas1: 100, Darbas2: 500, ...

### 2.2.5. Vartotojo sąsajos specifikacija

Vartotojo sąsajos reikalavimai: naudoti [PrimeNG](#) biblioteką su [Babylon](#) tema. Daugiau apribojimų nėra – šios sistemos pagrindinis akcentas nėra vartotojo sąsajos išvaizda.

## 2.2.6. Realizacijai keliami reikalavimai.

Realizacijos reikalavimai:

- Pamiršus slaptažodį, sugeneruojama nuoroda ir išsiunčiama į naudotojo el. paštą. Kai naudotojas paspaudžia nuorodą, gali nusistatyti naują slaptažodį.
- Sistemos naudotojams aprašyti apribojimai turi būti įgyvendinti ne tik vartotojo sąsajoje, bet ir serveryje.
- Sistema privalo turėti dokumentaciją naujam programuotojui, kurioje aprašyta, kaip paruošti savo kompiuterio aplinką darbui.
- Sistema privalo tikrinti savo kuriame programinį kodą su vienetų testais.
- Serveris turi būti paruoštas aptarnauti ne tik interneto svetainę, bet ir darbalaukio programą.

## 2.2.7. Techninė specifikacija.

Duomenų bazės valdymo sistema: PostgreSQL

Serverio programavimo kalba: C#

Serverio programavimo karkasas: ASP.NET (versija nuolat atnaujinama)

Serverio vienetų testų karkasas: xUnit

Serverio duomenų bazės valdymo įrankis: Entity Framework

Serverio saugumo pagalbinė sistema: Identity Server

Serverio kliento programavimo kalba: TypeScript

Serverio kliento programavimo karkasas: Angular (versija nuolat atnaujinama)

Serverio kliento komponentų biblioteka: [PrimeNG](#)

Failų saugykla: Azure Blob Storage.

## 2.3. Projektavimo eiga ir sistemos projektas

### 2.3.1. Projektavimo valdymas ir eiga

Prieš sistemos realizacijos pradžią, klientai pateikė xlsx tipo failą, kuriame buvo išvardinti visi funkciniai reikalavimai.

Kai projekto realizacijos eigoje atsiranda poreikis, klientai nubraižo būsenų diagramą, kuri programuotojų komandai atskleidžia visą objekto gyvavimo eigą.

### 2.3.2. Projektavimo technologija

Funkciniai reikalavimai buvo išvardinti su Microsoft „Excel“ įrankiu.

Būsenų diagramoms braižyti buvo naudojamas Microsoft „Visio“ įrankis.

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

Sistemos naudotojo sąsaja yra realizuojama su TypeScript, CSS bei HTML kalbų kombinacija. Taip pat naudojamos Angular karkaso teikiamais privalumais. Naudojamas kodo redaktorius: Visual Studio Code bei jo plati plėtinių aibė.

Sistemos serverio funkcionalumas yra programuojamas C# kalba, su .NET karkaso naujausia versija. Naudojama integruota programavimo aplinka: Visual Studio su Resharper plėtiniu.

Naudojama versijavimo kontrolės sistema: Git.

Naudojama operacinė sistema: Windows 10.

### 2.3.4. Informacijos srautai

Sistemos informacijos srautų diagrama pateikta 4 pav (diagrama yra supaprastinta siekiant apsaugoti kliento pranašumą rinkoje).



4 pav. Informacijos srautų diagrama

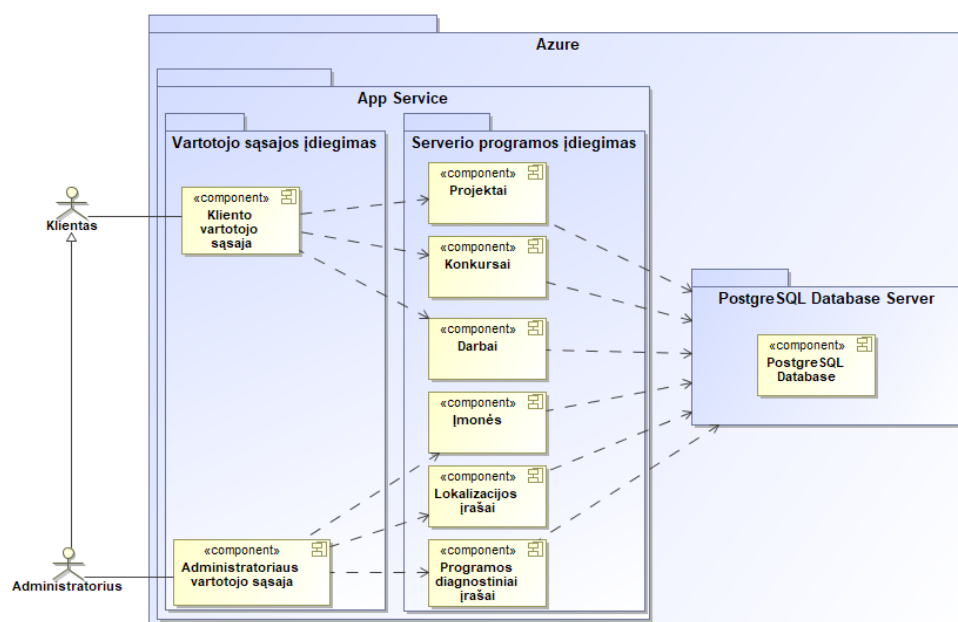
### 2.3.5. Sistemos sudėtis

Kuriamą sistemą sudaro trys pagrindinės dalys:

1. Duomenų bazė, kurioje yra saugoma visa statinė informacija.
2. Serveris, kuris turi tiesioginę prieigą prie duomenų bazės ir gali atlikti tarpinius veiksmus, jei tokių reikia.
3. Vartotojo sąsaja, per kurią vartotojas mato visą reikiamą informaciją bei gali daryti užklausas į serverį.

### 2.3.6. Komponentai

Sistemos komponentų diagrama pateikta 5 pav (diagrama yra supaprastinta siekiant apsaugoti kliento pranašumą rinkoje).



5 pav. Komponentų diagrama

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

Dabartiniame projekto realizavimo etape labiau rūpinamės baziniu funkcionalumu negu vartotojo sąsajos patyrimu. Kaip to pasekmė, vartotojo sąsaja naujiems vartotojams gali būti gana nepatogi. Visų pirma, kai kurios bazinių objektų formos yra apkrautos labai dideliu informacijos kiekiu, kuris gali eiti net per kelias korteles. Visų antra, kai kurie mygtukai neturi pavadinimų – tik ikonas. Kartais galima nesuprasti, ką tam tikra ikona reiškia.

Šiuo metu sistema pritaikyta jau patyrusiems vartotojams, kurie žino, kaip tokiomis sistemomis naudotis. Visgi, ateityje yra numatyta vartotojo patyrimo pusę patobulinti.



### 2.3.8. Duomenų kontrolė

Kiekvienas veiksmas yra validuojamas serveryje ir, jeigu įmanoma, vartotojo sąsajoje. Jei tam tikrų duomenų trūksta, arba įvesti duomenys yra nevalidūs, vartotojui parodomas pranešimas, kuriame tiksliai parašyta, kas negerai.

Duomenų kontrolės pavyzdys:

Jeigu vartotojas mėgins sukurti projektą neįrašęs jam jokio pavadinimo, sistema jau vartotojo sąsajos lygmenyje jam išmes klaidą – raudonai paryškins pavadinimo laukelį bei raudonomis raidėmis parašys žinutę „Pavadinimas yra privalomas“.

### 2.4. Testavimo eiga ir rezultatai

Šiame projekto realizacijos etape atliekame tik vienetų testavimą. Testuojame visus servisų viešus metodus, stengiamės ištestuos kiek įmanoma daugiau kelių.

Vienetų testavimui naudojama „xUnit“ biblioteka, pritaikyta C# kalba parašytoms programoms.

Vietoj reliacinės duomenų bazės, naudojame „Entity Framework“ bibliotekos suteiktą „In Memory Database“ alternatyvą, kuri imituoja reliacinę duomenų bazę darbinėje atmintyje.

Taip pat aktyviai naudojamos „MoQ“ biblioteka, kuri leidžia imituoti bet kokią sąsają – pavyzdžiui, jeigu servisas reikalauja el. laiškų siuntimo serviso, tačiau mes nenorime siųsti tikro el. laiško kiekvieną kartą, kai vykdomė vienetų testus.

Šiuo metu sistemai parašyti maždaug 800 vienetų testų.

### 2.5. Dokumentacija

Dokumentacija programuotojui:

1. Įdiegti Visual Studio Code.
2. Įdiegti NodeJS.
3. Įdiegti NPM CLI.
4. Įdiegti .NET 5.0 SDK,
5. Įdiegti PostgreSQL duomenų bazės serverį.
6. Susikonfigūruoti Microsoft User Secrets.
7. Sukompiliuoti programą.
8. Paleisti programą.

Dokumentacija vartotojui:

1. Nueiti į svetainę (adreso neminėsiu dėl konfidencialumo).
2. Užsiregistruoti per rekvizitus.
3. Prisijungti.
4. Prisijungus galima patekti į bet kurį klientui skirtą puslapį.

## Išvados

Praktika buvo atlikta sėkmingai, nes buvo pasiekti visi užsibrėžti rezultatai, kurie yra apibrėžti praktikos programoje:

- Sistemos klientas gali manipuluoti dokumentais.
- Sistemos klientas gali priimti svetimą konkursą.
- Sistemos klientas gali paskelbti kainos pasiūlymą svetimam konkursui.
- Sistemos klientas gali keisti puslapiuose naudojamo teksto kalbą.
- Sistemos klientas gali pakviesti svetimą įmonę prisijungti prie sistemos.

Visi tikslai buvo pasiekti iki užsibrėžtos datos – vėlavimų nebuvo.

Projektas turėjo tik kelias būsenos diagramas, kitų projektavimo modelių nebuvo. Visos ataskaitoje pateiktos diagramos buvo nubraižytos praktiką atlikusio studento.

Gauta kritika iš įmonės praktikos vadovo motyvavo tobulėti – supratau, kad skiriu per didelį dėmesį smulkmenoms, kas dažnai trukdo pastebėti didesnio masto problemas bei bereikalingai iššvaisto daug laiko.

Gauti pagyrimai taip pat buvo naudingi – teisingai atliekami darbo aspektai buvo užakcentuoti ir nepamiršti, todėl tų darbo aspektų nekeičiau, o atvirkščiai – dar labiau patobulinau.

Projekto komandos programuotojai dirbo su visais programos aspektais – nesikonzentruavo tik į vieną sritį. Tai padėjo pagilinti žinias visose srityse – tiek vartotojo sąsajos programavime, tiek serverio programavime, tiek reliacinėse duomenų bazėse, t.t. Kaip to pasekmė, pavyko pagilinti žinias visuose internetinės svetainės programavimo aspektuose. Kadangi praktikos rezultatai suteikė tiek daug naudos, darau išvadą, kad praktika vyko sklandžiai.

## Literatūra

1. Aufait Technologies (2021 m.). Nuskaityta 2021 m. kovo 15 d. iš „[Aufait](https://aufaittechnologies.com/tender-management-software) [1] – konkursų valdymo sistema, kuri skirta visiems.“: <https://aufaittechnologies.com/tender-management-software>
2. Odoo (2021 m.). Nuskaityta 2021 m. kovo 15 d. iš „[Odoo](https://www.odoo.com/) [2]– konkursų valdymo sistema, kuri yra skirta statybų įmonėms, tačiau visiškai nepritaikyta Lietuvos rinkai.“: <https://www.odoo.com/>
3. Statybų konkursai (2021 m.). Nuskaityta 2021 m. kovo 15 d. iš „[Statybų konkursai](https://statybukonkursai.lt/) [3]– konkursų valdymo sistema, kuri skirta statybų įmonėms ir pritaikyta Lietuvos rinkai“: <https://statybukonkursai.lt/>
4. Agile Documentation (2021 m.). Nuskaityta 2021 m. kovo 15 d. iš „Projektai yra vykdomi remiantis Agile [1] principais“: <https://docs.microsoft.com/en-us/azure/devops/boards/work-items/guidance/agile-process>