**Vilniaus Užupio gimnazija**

**Elektroninis dienynas privatiems darželiams**

Darbą atliko Herkus Valenta

Darbo vadovas Vaidas Raižys

**Vilnius, 2020**

Turinys

[Įvadas 3](#_Toc43139690)

[1. Programos kūrimo eiga 4](#_Toc43139691)

[1.1 Prisijungimo sistema 5](#_Toc43139692)

[1.1.1 Duomenų bazė 5](#_Toc43139693)

[1.1.2 Sistemos kūrimas 6](#_Toc43139694)

[1.1.3 Dizainas 7](#_Toc43139695)

[1.2 Įrankių juosta 8](#_Toc43139696)

[1.3 Vaikai 9](#_Toc43139697)

[1.3.1 Sistemos kūrimas 9](#_Toc43139698)

[1.4 Darbuotojai 10](#_Toc43139699)

[1.5 Lankomumas 10](#_Toc43139700)

[1.5.1 Sistemos kūrimas 11](#_Toc43139701)

[1.6 Atostogos 12](#_Toc43139702)

[1.7 Mokesčiai 12](#_Toc43139703)

[1.8 Programos diegimas 13](#_Toc43139704)

[2. Galimos programos kūrimo rizikos 14](#_Toc43139705)

[3. Programos testavimas 15](#_Toc43139706)

[Išvados 16](#_Toc43139707)

[Šaltiniai 17](#_Toc43139708)

# Įvadas

Taikomoji programinė įranga — programų visuma, naudojama kompiuteriui valdyti ar jame esančiai informacijai apdoroti. Dažniausiai tokia įranga kuriama naudojant programavimo kalbas, o vėliau kompiliuojant ar interpretuojant parašytą kodą. Programinės įrangos tikslas — palengvinti darbą vartotojui bei mažinti jo laiko sąnaudas.

Nepaisant sparčios informacinių technologijų plėtros, programinės įrangos kūrimas yra palyginti nauja industrijos šaka, kurios praktikos nėra nusistovėjusios. Tiek kūrėjai, tiek užsakovai ieško sprendimų, kaip optimizuoti programinės įrangos kūrimo procesus, kad minimaliomis lėšomis būtų sukurtas kokybiškas, vartotojo poreikius atitinkantis produktas.

Šią programą pradėjau kurti matydamas sunkų ir varginantį privačių darželių administracinį darbą. Mano mama privataus darželio direktorė, tačiau pati žymi vaikų lankomumą, skaičiuoja mokesčius. Toks darbas labai varginantis ir reikalaujantis daug susikaupimo, kad nebūtų administracinių klaidų. Norėdamas padėti, nusprendžiau sukurti programą, kurioje visi darželio administraciniai darbai persikeltų į vieną vietą ir skaičiavimus atliktų kompiuteris.

**Tikslas**:

Sukurti stabiliai veikiančią taikomąją programą, kuri palengvintų darželių administracijos darbą, padėtų sumažinti laiko sąnaudas bei administracinių klaidų kiekį.

**Uždaviniai**:

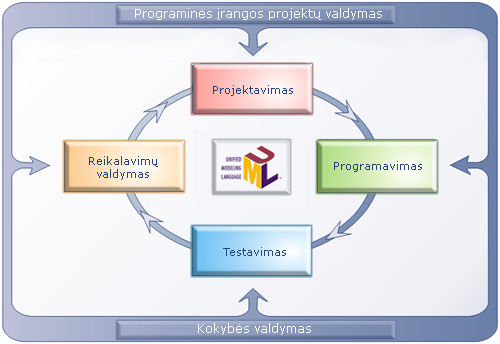
* Sukurti duomenų bazę, kuri leistų vartotojui prisijungti prie programos;
* Sukurti programos šabloną su pagrindinėmis funkcijomis :

1. Vaikų lankomumo sistema;
2. Mokesčių skaičiuoklė;
3. Vaikų ir darbuotojų informacijos saugykla;
4. Atostogų pasirinkimo sistema;

* Sukurti duomenų bazę, kurioje bus laikomi įstaigos duomenys, susieti ją su programos kodu;
* Sukurti vartotojui suprantamą ir patrauklią programos išvaizdą;

# 1. Programos kūrimo eiga

Programinės įrangos kūrimo veiklos dažnai skirstomos į inžinerines ir valdymo. Dažniausiai išskiriamos keturios pagrindinės inžinerinės veiklos: reikalavimų analizė, projektavimas, programavimas ir testavimas. Šios veiklos formaliai ar neformaliai atliekamos visuose programinės įrangos projektuose.

Taikomųjų programų kūrimas prasideda nuo reikalavimų aprašymo ir problemos nagrinėjimo. Labai svarbu suprasti, kad programinės įrangos projektų reikalavimai turi būti nuosekliai analizuojami keliuose lygiuose – nuo vizijos ir verslo tikslų iki galutinio vartotojo scenarijų bei realizacijos funkcijų aprašymo.

1 pav. Programinės įrangos kūrimo procesas

Taikomąją programą nusprendžiau kurti pasitelkiant programinės įrangos kūrimo aplikacija „Visual studio“ bei C# programavimo kalba. Su šia programavimo kalba jau esu susipažinęs, ją naudojau kuriant žaidimus, tad sintaksė ir funkcijų pavadinimai jau buvo žinomi.

## 1.1 Prisijungimo sistema

Prisijungimas — procesas kompiuterių saugumo srityje, kurio metu asmuo įgyja prieigą prie programinės įrangos ar kompiuterinės sistemos, save identifikuodamas. Prisijungimas dažniausiai yra tam tikras vartotojo vardas bei slaptažodis.

Šis procesas yra neišvengiama mano kuriamos programos dalis, nes sistema atskyrusi, kuris vartotojas prisijungė, suteikia prieiga tik prie jam priklausančios informacijos. Šio proceso įgyvendinimui turėjau naudoti duomenų bazę.

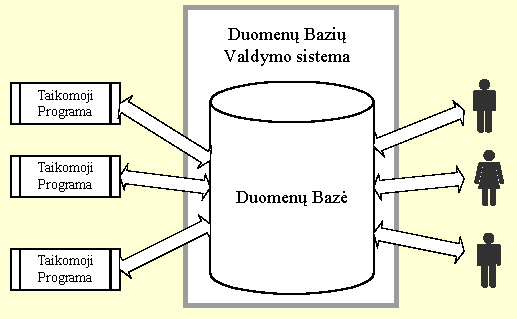
### 1.1.1 Duomenų bazė

Nūdienos pasaulyje dažnai duomenys yra kaupiami, saugomi ir apdorojami kompiuteriuose. Jeigu duomenys naudojami tiktai vienai problemai spręsti, jų parengimu, laikymu ir tvarkymu dažniausiai rūpinasi pats problemos sprendėjas. Tačiau kai tie patys duomenys naudojami kelių problemų sprendimui, tampa aktuali įvairių uždavinių poreikių suderinamumo problema. Akivaizdu, kad pastaruoju atveju duomenis reikia parengti taip, kad jie tiktų kuo didesniam uždavinių skaičiui. Tam naudojamos duomenų bazės.

Duomenų bazė — organizuotas bei struktūrizuotas informacijos ar duomenų saugojimas kompiuterinėje sistemoje. Duomenų bazei sukurti ir valdyti naudojama duomenų bazių valdymo sistema (DBVS) kuri turi savo atskira manipuliavimo kalbą.

Svarbiausios duomenų bazės valdymo sistemos funkcijos:

* Duomenų bazės struktūros projektavimas;
* Duomenų peržiūrėjimas, paieška, rikiavimas ir kitas tvarkymas;
* Taikomųjų vartotojo programų kūrimas;
* Ataskaitų kūrimas;

2 pav. Duomenų bazės veikimo principas

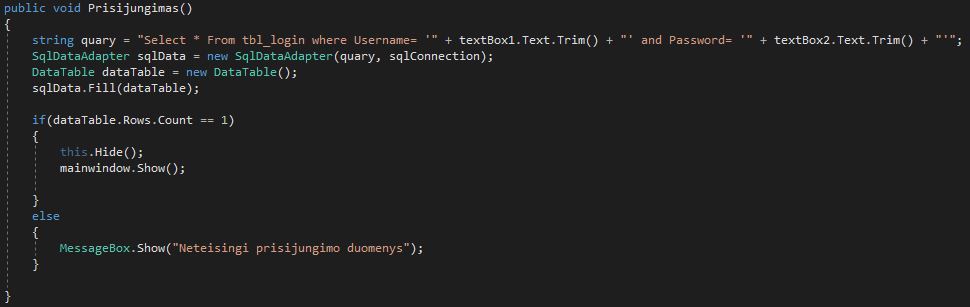
Duomenų bazės aprašymą reikia sudaryti taip, kad duomenų apdorojimo programos galėtų pačios atsirinkti iš laikomų duomenų joms reikalingus elementus, sukomponuoti iš jų naujus darinius ir panaudoti juos savo sprendžiamuose uždaviniuose.

### 1.1.2 Sistemos kūrimas

Norint pradėti darbą su duomenų baze, reikia turėti serverį, kuriame kaupiama informacija. Kuriant programą pasitelkiau „Microsoft SQL Server“ ir programavimo kalbą SQL, skirtą aprašyti duomenis ir jais manipuliuoti.

„Microsoft SQL Server“ — nemokama reliacinė duomenų bazių valdymo sistema sukurta Microsoft. Šios programinės įrangos dėka yra lengviau valdyti ir atkurti duomenis esančius serveryje bei juos valdyti pasitelkiant tinklą. Šios programos kodą galima susieti su daugeliu taikomųjų programų kūrimo aplikacijomis, taip pat ir su mano naudojama.

Funkcijos kūrimo etapai:

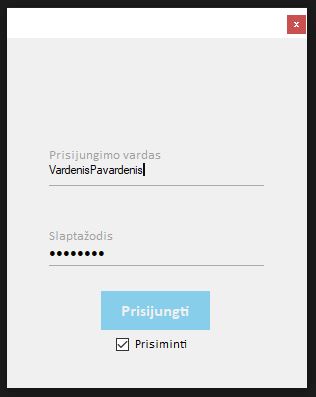
* Suvesti duomenis, kurie bus naudojami prisijungiant, į duomenų bazę;
* Duomenų bazę susieti su programos kodu nurodant serverio vietą ir kitus parametrus;
* Patikrinti ar duomenys teisingi;
* Jei duomenys teisingi — atverti pagrindinį programos langą, jei ne — pranešti, kad duomenys neteisingi.

3 pav. Prisijungimo funkcijos kodas

### 1.1.3 Dizainas

Prisijungimo lango dizainą sudaro:

* Pagrindinis langas;
* Teksto laukai — vartotojo duomenims įvesti;
* Prisijungimo mygtukas — funkcijai įvykdyti.
* Mygtukas „Prisiminti“ — vartotojo duomenų išsaugojimas greitesniam prisijungimui

4 pav. Prisijungimo langas

## 1.2 Įrankių juosta

Prisijungus prie programos atsidaro pagrindinis programos langas, kurio dešiniajame krašte yra įrankių juosta. Šioje juostoje esantys mygtukai nukreipia vartotoją į kitus programos langus. Juostos apačioje yra atsijungimo ir programos išjungimo mygtukai.

Įrankių juostos mygtukai:

* Lankomumas
* Vaikai
* Darbuotojai
* Atostogos
* Mokesčiai
* Nustatymai
* Išjungti
* Atsijungti

## 1.3 Vaikai

Viena iš programos funkcijų yra vaikų duomenų saugykla. Ši funkcija leidžia vartotojui pridėti naują vaiką į duomenų bazę, kurio duomenis galima peržvelgti, atnaujinti ar ištrinti. Vaikų duomenų saugykla — greitas ir efektyvus būdas rasti tam tikrą informaciją apie vaiką ar jo tėvų kontaktus. Norint įgyvendinti šią funkciją, teko naudoti kelias duomenų bazes bei automatiškai generuoti mygtukus.

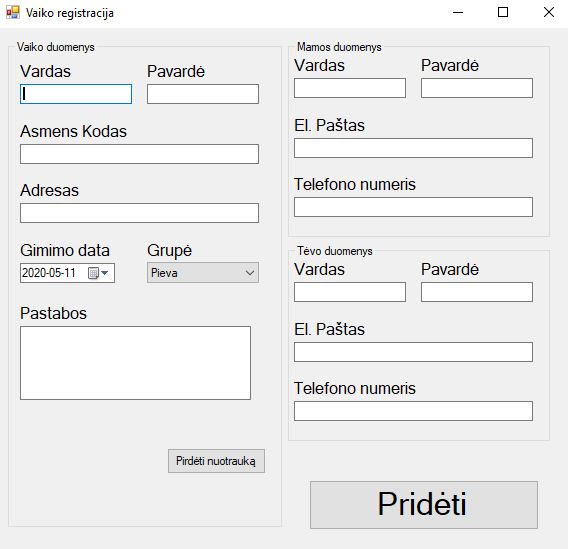
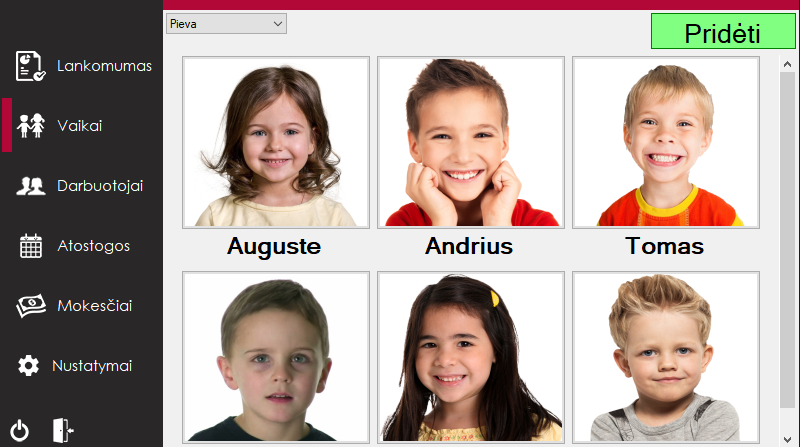
### 1.3.1 Sistemos kūrimas

Kuriant šios funkcijos kodą vadovavausi šiuo planu:

1. Sukurti atskirą programos langą, kuriame vartotojas gautų prieiga prie duomenų registravimo.
2. Sukurti naują duomenų bazę, kurioje bus talpinami vaikų duomenys.
3. Sukurti duomenų bazę, kurioje bus talpinami automatiškai, pagal vaikų kiekį, besigeneruojantys mygtukai.
4. Duomenų pridėjimo, atnaujinimo ir ištrynimo sistemos.
5. Susieti kodą su programos lange esančias mygtukais.

Pirmieji žingsniai buvo sukurti teksto laukus, kuriuose rašoma informacija. Šie laukai suskirstyti į tris grupes, tai yra vaiko, mamos ir tėvo duomenys. Paspaudus mygtuką „Pridėti“ iššaukiama funkcija, kurios dėka duomenys iš teksto laukų nuskaitomi ir perkeliami į duomenų bazę.

Kai tik duomenys yra išsaugomi duomenų bazėje, iššaukiama kita funkcija, kuri nuskaito, kiek vaikų yra pridėta ir naudoja žinomo kartojimų skaičiaus ciklą „FOR“, kurio metu ištraukiama informacija apie vaikus ir sugeneruojami mygtukai, kurie taip pat talpinami į duomenų bazę tolimesniam naudojimui. Vos tik atidarius programinį langą, programa nuskaito mygtukų duomenų bazę, sugeneruoja juos ir išdėsto programos lange. Paspaudus ant mygtuko atsidaro kitas langas, kuriame matoma informacija apie vaiką.

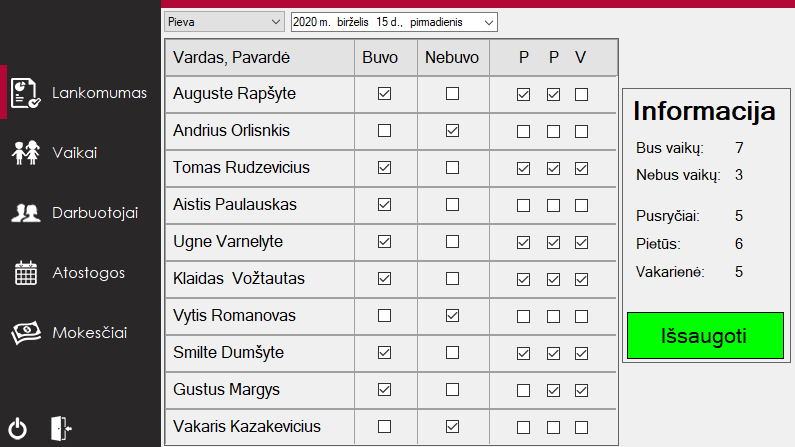
5 pav. Vaiko registracijos langas 6 pav. Programinis langas „Vaikai“

## 1.4 Darbuotojai

Ši programos skiltis labai panaši į vaikų duomenų saugyklą, tik šioje talpinama informacija yra apie darbuotojus. Darbuotojų duomenų saugykla — tik administracijai prieinamas programinis langas, kuriame talpinama svarbiausia informacija apie įmonėje dirbančius asmenius. Šios funkcijos dėka svarbūs duomenys, pavyzdžiui, kaip įdarbinimo data, medicininės pažymos galiojimo laiko pabaiga, randami daug greičiau ir vienoje vietoje. Programos kodas beveik nieko nesiskiria nuo vaikų duomenų saugyklos kodo, teko panaudoti dvi duomenų bazes ir automatiškai besigeneruojančius mygtukus.

## 1.5 Lankomumas

Pagrindinė programos funkcija yra vaikų lankomumo sistema. Lankomumo žymėjimas privatiems darželiams yra būtinas, kadangi mėnesinis mokestis yra apskaičiuojamas remiantis, kiek dienų vaikas buvo atvykęs bei kiek kartų per dieną valgė (jei šie mokesčiai nėra fiksuoti). Sistema leidžia darbuotojams sužymėti, kurie vaikai yra bei ką valgė (pusryčiai, pietūs bei vakarienė). Visi duomenys išsaugomi duomenų bazėse, tad administracijai nereikia vargti su popierizmu ir viską gali valdyti vienoje platformoje.

7 pav. Programinis langas „Lankomumas“

### 1.5.1 Sistemos kūrimas

Šios sistemos kūrimas man buvo didžiausias iššūkis kuriant šią programą. Teko daug svarstyti, kaip išpildyti duomenų bazių veiksmingumą. Vaikų lankomumas dažnai surašomas į ataskaitas, tad svarbu, kad duomenys būtų išsaugomi ilgam laikui. Tad didžiausias keblumas buvo nuspręsti, kaip išspręsti duomenų bazės suderinamumo problemą. Po ilgo galvos sukimo ir daug bandymų supratau, kad geriausias būdas, kaip realizuoti suderinamumą, yra sukurti mėnesines duomenų bazes.

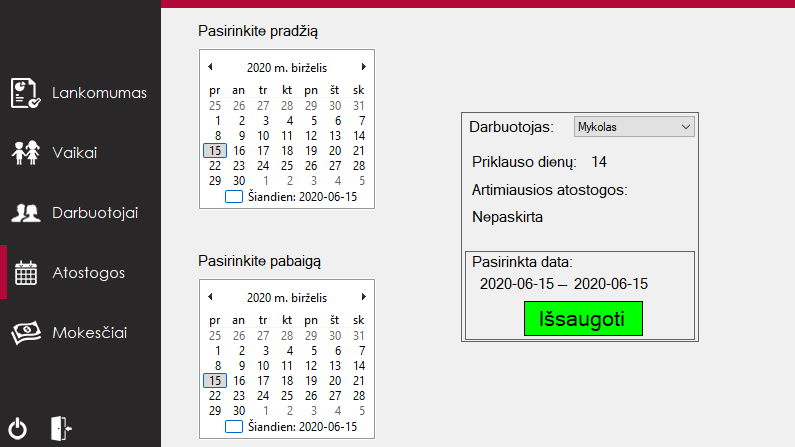
Kiekvienais metais reikia sukurti 12 skirtingų duomenų bazių. Duomenys bus suskirstyti tokia tvarka:

1. Vaiko vardas, pavardė.
2. Vaiko grupė.
3. Įrašymo data.
4. Lankomumo duomenys.

Kai lankomumo programiniame lange yra paspaudžiamas mygtukas „Išsaugoti“, duomenys yra perkeliami į duomenų bazę ir yra paruošti tolimesniam naudojimui. Kai norima peržvelgti dienos ar mėnesio ataskaitą, atsižvelgiama į įrašymo datos informaciją. Pavyzdžiui, jeigu reikia peržiūrėti Liepos 12 dienos ataskaitą, tuomet duomenų bazėje ieškomi visi įrašai atitinkantys šį rodiklį. Kai žinomi visi objektai pagal datą, jie rūšiuojami pagal grupes. Turint visą reikiamą informaciją yra sugeneruojama lentelė, kurioje surašyti tos dienos duomenys.

## 1.6 Atostogos

Atostogų pasirinkimo sistema — efektyvus būdas paskirstyti atostogas įstaigos darbuotojams. Šioje programos skiltyje darbdavys gali matyti, kokiam laikotarpiui gali išleisti darbuotojus, jo laisvas dienas. Kalendoriuje rodoma kurios dienos yra užimtos ir pagal tai gali paskirti, kada darbuotojas gali išeiti atostogų.

Sistemai sukurti reikėjo naudoti duomenų bazę, kurioje talpinami duomenys apie darbuotojų laisvas dienas. Programiniame lange yra du mėnesiniai kalendoriai, kurie leidžia nustatyti dienų intervalą, kuriam reikia išleisti žmogų. Pasirinkus intervalą nuskaitoma ar darbuotojui turi pakankamai dienų atostogauti tokiam laikotarpiui.

8 pav. Programinis langas „Atostogos“

## 1.7 Mokesčiai

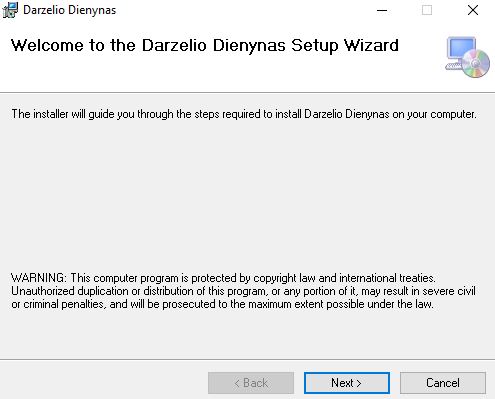
Viena iš svarbiausių šios programos funkcijų yra mokesčių skaičiuoklė. Šios funkcijos tikslas — atsižvelgti į lankomumo grafiką ir pateikti kiekvieno vaiko skirtingą mėnesinį mokestį. Kiekvieną mėnesį administracijai reikia peržvelgti lankomumo ataskaitas, suskaičiuot vaiko lankytas dienas, maisto porcijas per dieną ir tuomet apskaičiuoti galutinį mokestį. Toks darbas yra labai varginantis bei labai atsakingas ir dažnai gali pasitaikyti administracinių klaidų. Tad programos dėka šis darbas bus atliekamas kompiuteriu, vieno mygtuko paspaudimu.

Skaičiuoklė veikia labai paprastai. Paspaudus mygtuką „Apskaičiuoti“, iš lankomumo duomenų bazių ištraukiama informacija apie pasirinktą vaiką. Skaičiuojant mokestį atsižvelgiama į tai, kiek dienų vaikas buvo darželyje ir kiek kartų valgė. Šie rodikliai susumuojami ir pateikiami galutinė kaina.

9 pav. Greito prototipo metodas

## 1.8 Programos diegimas

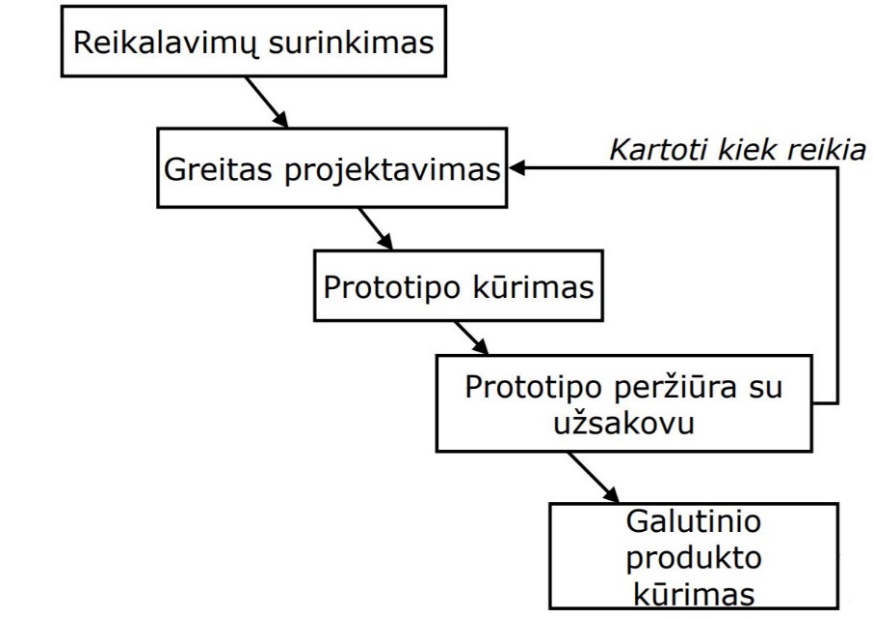
Programos diegimas — procesas, kurio metu taikomoji programa yra įrašoma į vartotojo kompiuterį. Elektroninio dienyno diegimo metu yra įrašomi pagrindiniai programos failai ir lokalus Microsoft SQL serveris, kuris suteikia prieiga prie duomenų bazių. Diegimo procesas labai paprastas, reikalaujantis tik nustatyti kuriame diske įrašyti programą. Diegimo lango kūrimas buvo nesudėtingas. Pasitelkiant „Visual studio“ programų diegimo papildinį reikėjo nurodyti kuriuos failus ir kokias papildomas programas reikės įdiegti.

10 pav. Diegimo langas

# 2. Galimos programos kūrimo rizikos

Šią programą kūriau naudodamas greito prototipo metodą. Šis metodas naudojamas tada, kai programos pagrindą sudaro kliento norai ir idėjos. Vadovaujantis šiuo principu vartotojas yra visuomet įsitraukęs į sistemos kūrimą, taip jis labiau pasitikės sistema ir ji atitiks jo reikalavimus. Greito prototipo kūrimo metodas tinkamas, kai reikalavimai nėra patvirtinti ar nuolat kintantys. Tai padeda mažinti programinės įrangos kūrimo rizika, gerinti kokybę bei našumą.

Tačiau vadovaujantis šiuo metodu gali kilti daug kūrimo rizikos problemų. Dažniausiai nestabilus ar prastai realizuotas prototipas gali virsti galutiniu produktu. Taip pat reikalaujama nuolatinio ir intensyvaus užsakavo bendradarbiavimo, kuris gali varginti vartotoją ar atitraukti jį nuo darbų.



11 pav. Greito prototipo metodas

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Rizikos faktorius** | **Tikimybė** | **Įtaka** |
|  | Projekto nutraukimas | Labai maža | Neleistina |
|  | Programos dizaino pakeitimas | Vidutinė | Rimta |
|  | Prastai realizuotas prototipas — galutinis produktas | Didelė | Vidutinė |
|  | Projekto vėlavimas | Didelė | Rimta |

lentelė. Rizikos faktorių sąrašas.

# 3. Programos testavimas

Programinės įrangos be defektų nebūna. Norint užtikrinti, kad pasiekti rezultatai tenkina lūkesčius reikalingas galutinio produkto testavimas. Labai svarbu, kad programinę įrangą, programinį kodą vertintų ne kūrėjas, o kitas žmogus. Jei programinė įranga yra vertinama autoriaus, dažniausiai defektai būna nepastebėti.

Pastebėtas spragas ištašyti akimirksniu yra neįmanoma, tad visi defektai yra užregistruojami. Šios klaidos skirstomos pagal jų prioritetą, kuris nurodo jo žalą programos naudojimui ir kaip dažnai defektas pasitaiko. Kiekvienas programinėje įrangoje pasitaikęs defektas turi būti pašalintas, tačiau tai nereiškia jo ištaisymą, kai kurie defektai gali būti atidedami arba nepataisomi, tai yra ištrinama programos dalis. Jei problema išspręsta, reikia patikrinti, ar ji pašalinta ir ar nesąlygojo naujų defektų atsiradimo.

Norint pateikti programą vartotojui, reikia įsitikinti, kad pašalinti aukščiausio prioriteto bei didžiausio žalingumo defektai ir yra užtikrintas aukščiausias kokybės lygis. Mano sukurtos programos testavimui suplanuotas trijų mėnesių testavimas, kurio metu bus ieškoma programos kode esančių trūkumų, reikalavimų nesuderinamumų ir duomenų bazių veiklos trukdžių.

Išvados

Visi uždaviniai įgyvendinti. Programa veikia sklandžiai, suprantama vartotojui. Šios programos dėka darželio administracijai bus lengviau dirbti, pagerės darbo kokybė, bus sutaupoma laiko bei sumažės administracinių klaidų rizika.

Šis projektas įtvirtino mano žinias C# programavimo kalboje, pagilinau „Visual studio“ naudojimo žinias ir išmokau dirbti su duomenų bazėmis. Šis projektas buvo naudingas man, nes galiu jį įtraukti į darbų portfolio, puikiai atspindi mano įgytą patirtį bei sukauptas žinias. Tai gali padėti gauti geresnių, naujų užsakymų ar darbo pasiūlymų.

Šaltiniai

*Duomenų bazės* [interaktyvus]. since loco, s.l., b.v. [žiūrėta 2020 m. gegužės 4 d.]. Prieiga per internetą: <https://www.paltarokogimnazija.lt/sena/Informatika/access20004_001.htm>

*Pagrindinis* [interaktyvus]. since loco, s.l., b.v. [žiūrėta 2020 m. balandžio 27 d.]. Prieiga per internetą: <https://mlukas.weebly.com/>

ŠILINGAS, Darius *Programinės įrangos kūrimas* [interaktyvus] [žiūrėta 2020 m. balandžio 27 d.]. Prieiga per internetą: <http://www.nomagic.lt/straipsniai/programines-%C4%AFrangos-kurimas.html>

Computer Hope *Login* [interaktyvus] [žiūrėta 2020 m. balandžio 27 d.]. Prieiga per internetą: <https://www.computerhope.com/jargon/l/login.htm>