



Kauno technologijos universitetas

Informatikos fakultetas

Vartotojo sąsajų prototipų generavimas pasitelkiant dirbtinį intelektą

Baigiamasis magistro projektas

IFM-9/2 gr. Ieva Jakšaitytė

Projekto autorė

Prof. Tomas Blažauskas

Vadovas

Kaunas, 2021



Kauno technologijos universitetas

Informatikos fakultetas

Vartotojo sąsajų prototipų generavimas pasitelkiant dirbtinį intelektą

Baigiamasis magistro projektas

Programų sistemų inžinerija (6211BX011)

IFM-9/2 gr. Ieva Jakšaitytė

Projekto autorė

Prof. Tomas Blažauskas

Vadovas

Doc. Šarūnas Packevičius

Recenzentas

Kaunas, 2021



Kauno technologijos universitetas

Informatikos fakultetas

Ieva Jakšaitytė

Vartotojo sąsajų prototipų generavimas pasitelkiant dirbtinį intelektą

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Ieva Jakšaitytė

Patvirtinta elektroniniu būdu



Kauno technologijos universitetas

Informatikos fakultetas

Baigiamojo magistro projekto užduotis

Projekto tema Vartotojo sąsajų prototipų generavimas pasitelkiant dirbtinį intelektą

Reikalavimai ir sąlygos
(tikslinti pavadinimą
pagal poreikį)

Vadovas / Vadovė

Prof. Tomas Blažauskas

2021-05-17

(vadovo pareigos, vardas, pavardė, parašas)

(data)

Jakšaitytė Ieva. Vartotojo sąsajų prototipų generavimas pasitelkiant dirbtinį intelektą. Magistro baigiamasis projektas / vadovas prof. Tomas Blažauskas; Kauno technologijos universitetas, informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): programų sistemos, informatikos inžinerija.

Reikšminiai žodžiai: grafinė vartotojo sąsaja; dirbtinis intelektas; mašininis mokymasis; vartotojo sąsajos prototipas; eskizas; eskizo atpažinimas; prototipavimas; prototipo generavimas.

Kaunas, 2021. 109 p.

Santrauka

Šiuo metu projektuojant vartotojo sąsają galima naudoti įvairius projektavimo būdus ir formatus. Dažnai vartotojo sąsajos projektuotojo ar dizainerio užduotis yra suprojektuoti vartotojo sąsają nuo idėjos iki susitarto lygio vartotojo sąsajos prototipo ar dizaino. Žmonėms patogiau kai naujas idėjas jie gali pavaizduoti piešiant eskizus ant popieriaus lapo, tačiau tokie eskizai gali perteikti tik bendrinius vartotojo sąsajos aspektus. Todėl norint turėti daugiau detalių ir galimybę geriau ištestuoti sugalvotas idėjas, reikia piešti vartotojo sąsajos prototipą kompiuterio pagalba. Tačiau perpiešti eskizus į prototipą užtrunka daugiau laiko, tai yra labai monotoniškas darbas bei yra be reikalo švaistomas projektuotojo laikas, kurį jis galėtų panaudoti svarbesnėms užduotims atlikti.

Norint išspręsti šią problemą būtų galima sukurti sprendimą, kuris iš ant popieriaus nupiešto eskizo suprastų jame pateiktą žymėjimą bei nupieštų prototipą pats. Tokiu būdu projektuotojas galėtų skirti daugiau laiko idėjų generavimui ir vartotojo sąsajos testavimui.

Šiame darbe buvo analizuoti ir tyrinėjami kiti literatūra ir kiti panašūs sprendimai. Atsižvelgiant į problemą, potencialius sistemos naudotojus, galimus sprendimo kelius, įvairius apribojimus ir kitas detales, buvo suprojektuota ir realizuota sistema. Sukurtos sistemos patogumas, būsimiems sistemos naudotojams, buvo ištestuotas pasitelkiant tinkamumo naudoti testavimą. Pavykus gauti pirmuosius rezultatus eskizų atpažinime ir rezultatų sugeneravime, buvo sukurta daugiau ir įvairesnių duomenų, kurie buvo panaudoti apmokant modelį naujų žymėjimų eskizuose.

Jakšaitytė Ieva. Generating User Interface Prototypes Using Artificial Intelligence. Master's Final Degree Project / supervisor prof. Tomas Blažauskas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): software engineering, informatics engineering.

Keywords: graphical user interface; artificial intelligence; machine learning; user interface prototype; sketch; sketch recognition; prototyping; prototype generation.

Kaunas, 2021. 109 p.

Summary

Currently, a variety of design methods and formats can be used to design the user interface. Often, the task of a user interface designer or designer is to design a user interface from an idea to an agreed-upon level of user interface prototype or design. People find it most convenient to come up with new ideas by drawing sketches on a piece of paper, but such sketches can only convey general aspects of the user interface. Therefore, in order to have more details and the opportunity to better test the ideas, it is necessary to draw a prototype of the user interface with the help of a computer. However, repainting sketches into a prototype takes more time, which is a very monotonous job and is an unnecessarily wasted time for the designer, which he could use for more important tasks.

To solve this problem, it would be possible to create a solution that would understand the notation in the sketch drawn on the paper and draw the prototype itself. In this way, the designer could spend more time generating ideas and testing the user interface.

In this work, other literature and other similar solutions were analysed and researched. Depending on the problem, potential system users, possible solution paths, various limitations and other details, the system was designed and implemented. The convenience of the system for future users, was tested through usability testing. After the first results in sketch recognition and result generation were obtained, more diverse data were generated that were used to train the model in the sketches of the new markings.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	9
Santrumpų ir terminų sąrašas	11
Įvadas.....	12
1. Analitinė dalis	14
1.1. Įvadas.....	14
1.2. Srities apžvalga.....	14
1.2.1. Vartotojo sąsajos prototipavimas	14
1.2.2. Pirminis vaizdų apdorojimas	17
1.2.3. Iššūčiai eskizų nuskaityme, modelio atpažinime	20
1.2.4. Kompiuterinio matymo technikos (angl. <i>computer vision techniques</i>).....	21
1.2.5. Mašininio mokymosi technikos.....	21
1.2.6. Kodo generavimas	23
1.3. Apibendrinančios išvados.....	24
2. Projektinė dalis	25
2.1. Duomenų modelis ir jo elementų žodynas	25
2.1.1. Duomenų modelis.....	25
2.1.2. Duomenų žodynas (duomenų modelio specifikacija)	25
2.2. Sistemos panaudos atvejų diagrama	27
2.3. Nefunkciniai reikalavimai	29
2.4. Sistemą sudarantys paketai.....	32
2.5. Sistemos diegimo vaizdas.....	35
2.6. Duomenų vaizdas	36
3. Tyrimo dalis	37
3.1. Išplėsto modelio tyrimas.....	37
3.1.1. Projekto dirbtinio intelekto dalies apžvalga	37
3.1.2. Duomenų paruošimas modelio apmokymui.....	42
3.1.3. Projekto paruošimas modelio apmokymui ir galutinių rezultatų sugeneravimui.....	47
3.2. Tinkamumo naudoti testavimas (angl. <i>usability testing</i>).....	48
3.3. Tinkamumo naudoti sistemą tyrimas.....	50
4. Eksperimentinė dalis	53
4.1. Išplėsto modelio tyrimas.....	53
4.2. Tinkamumo naudoti testavimas (angl. <i>usability testing</i>).....	57
4.3. Tinkamumo naudoti sistemą tyrimas.....	61
Išvados	64
Literatūros sąrašas	65
Priedai.....	68
1 priedas. Duomenų modelio specifikacijos.....	68
2 priedas. Panaudos atvejų specifikacijos	70
3 priedas. Nefunkciniai reikalavimai	85
4 priedas. Detalizuotos paketų diagramos	89
5 priedas. Tinkamumo naudoti testavimo scenarijus	99
6 priedas. Tinkamumo naudoti testavimo užrašai	102

Lentelių sąrašas

2.1 lentelė. Duomenų modelio „Projektas“ klasės ir jos atributų aprašymas.....	26
2.2 lentelė. Duomenų modelio „Prototipas“ klasės ir jos atributų aprašymas	26
2.3 lentelė. Duomenų modelio „Eskizas“ klasės ir jos atributų aprašymas	26
2.4 lentelė. Panaudos atvejo „Sugeneruoti prototipo langą iš eskizo nuotraukos“ specifikacija	28
2.5 lentelė. Panaudos atvejo „Parsisiųsti prototipą“ specifikacija	29
2.6 lentelė. Reikalavimas sistemos išvaizdai (išvaizdos reikalavimas)	29
2.7 lentelė. Reikalavimas panaudojamumui (naudojimosi paprastumas)	30
2.8 lentelė. Reikalavimas panaudojamumui (mokymosi reikalavimas).....	30
2.9 lentelė. Reikalavimas efektyvumui ir našumui (išplečiamumo reikalavimas).....	30
2.10 lentelė. Pritaikymo kitoms platformoms (angl. <i>portability</i>) reikalavimas	31
3.1 lentelė. Pradinių duomenų rinkinio eskizų ir naujai sukurtų eskizų palyginimas.....	46
4.1 lentelė. Tinkamumo naudoti testavimo metu surastos sistemos klaidos.....	58
4.2 lentelė. Dėl tinkamumo naudoti testavimo rezultatų siūlomos tobulintinos vietos	59
4.3 lentelė. Tinkamumo naudoti klausimyno rezultatai	62
4.4 lentelė. Tinkamumo naudoti įverčio skaičiavimai	62

Paveikslų sąrašas

1.1 pav. Popierinio, kompiuterinio ir veikiančio prototipų palyginimas [2].....	15
1.2 pav. Mažo sudėtingumo lygio prototipas [12]	16
1.3 pav. Vidutinio sudėtingumo lygio prototipas [12]	16
1.4 pav. Didelio sudėtingumo lygio prototipas [12]	16
1.5 pav. Linijų šalinimas apdorojant vaizdą [13].....	18
1.6 pav. Linijų tiesinimas ir gražinimas apdorojant vaizdą [23].....	18
1.7 pav. Atvaizdas su netvarkinga perspektyva [17].....	19
1.8 pav. Linijų aptvarkymas vaizde [16].....	19
1.9 pav. Linijų aptvarkymas vaizde [18].....	19
1.10 pav. (a) pradinis vaizdas, (b) linijų kampų šalinimas, (c) linijų normalizacija [21]	20
1.11 pav. Semantinio segmentavimo pavyzdys [32].....	23
1.12 pav. <i>pix2code</i> modelio architektūra [33].....	24
2.1 pav. Duomenų modelio diagrama	25
2.2 pav. Panaudos atvejų diagrama	27
2.3 pav. Sistemą sudarantys paketai.....	32
2.4 pav. Paketo <i>API::Models</i> klasių diagrama	33
2.5 pav. Paketo <i>API::Controllers</i> klasių diagrama	34
2.6 pav. Sistemos diegimo vaizdas	35
2.7 pav. Duomenų bazės modelio schema	36
3.1 pav. Dirbtinio intelekto modelio failas.....	37
3.2 pav. Originalaus duomenų rinkinio eskizo nuotrauka.....	38
3.3 pav. Originalaus duomenų rinkinio eskizo aprašas.....	38
3.4 pav. Originalaus duomenų rinkinio žodynas.....	39
3.5 pav. Originalaus duomenų rinkinio transformavimo į saityno programą žodynas	39
3.6 pav. Modelio apmokymo diagrama	40
3.7 pav. Eskize esančios informacijos atpažinimo diagrama.....	40
3.8 pav. Eskizo aprašo transformavimo į <i>*.html</i> diagrama	41
3.9 pav. Atpažinto eskizo tikslumo nustatymas	41
3.10 pav. Stulpelio ir jo turinio generavimas <i>*.gui</i> ir <i>*.html</i> formatais	42
3.11 pav. Sugeneruoto eskizo informacijos įrašymas į failus ir paveiksluko sukūrimas.....	43
3.12 pav. Duomenų (eskizų) generavimas	43
3.13 pav. Naujai sukurto eskizo pavyzdys (1)	44
3.14 pav. Naujai sukurto eskizo pavyzdys (2)	45
3.15 pav. Naujai sukurto eskizo pavyzdys (3)	46
3.16 pav. Naujai sukurto eskizų rinkinio žodynas	47
3.17 pav. Eskizo aprašo transformavimo į kitą formatą žodyno failas	48
3.18 pav. Įverčiai, tinkamumas naudoti, priėmimas ir NPS kategorijos, susijusios su SUS įverčiais [39]	51
3.19 pav. SUS grafike kartu su procentiniu įvertinimu ir balais [39]	52
3.20 pav. Tinkamumo naudoti įvertinimas (angl. <i>System Usability Scale (SUS)</i>), priimtinumas	52
4.1 pav. Eskizų generavimo paleidimas.....	54
4.2 pav. Netinkamo eskizo dėl išlendančio turinio (kairėje) ir tinkamo eskizo (dešinėje) pavyzdžiai	54
4.3 pav. Eskizų validavimas ir netinkamų šalinimas	55

4.4 pav. Duomenų rinkinio išskaidymas į treniravimo ir validavimo rinkinius su <i>pix2code</i>	55
4.5 pav. Eskizų nuotraukų pavertimas į <i>numpy</i> bibliotekos masyvus su <i>pix2code</i>	55
4.6 pav. Modelio apmokymas su <i>pix2code</i>	56
4.7 pav. Eskizų aprašų (*. <i>gui</i>) generavimas iš eskizų nuotraukų su <i>pix2code</i>	56
4.8 pav. Eskizo aprašo (*. <i>gui</i>) generavimas iš eskizo nuotraukos su <i>pix2code</i>	56
4.9 pav. BLEU įverčio skaičiavimas su <i>SketchCode</i>	56
4.10 pav. Originalus eskizo aprašas (kairėje) ir nuspėtas eskizo aprašas (dešinėje).....	57
4.11 pav. Tinkamumo naudoti testavimo papildomų klausimų rezultatai	61
4.12 pav. Sistemos tinkamumo naudoti įvertis kartu su procentiniu įverčiu	63
4.13 pav. Išmokstamumo (angl. <i>learnability</i>) ir tinkamumo naudoti (angl. <i>usability</i>) įverčiai pagal dalyvius.....	63

Santrumpų ir terminų sąrašas

Terminai:

Eskizas (angl. *sketch*) – grubiai, ant popieriaus lapo, nupieštas dizainas, pavyzdžiui, internetinio tinklalapio naudotojo sąsajos.

Skaitmeninis eskizas (angl. *wireframe*) – kompiuteryje nupiešti internetinio tinklalapio naudotojo sąsajos „griaučiai“.

Prototipas (angl. *prototype*) – ankstyvas produkto pavyzdys, modelis ar leidimas, pagamintas produkto koncepcijos ar proceso testavimui.

Prototipo sudėtingumo (detalizavimo) lygis (angl. *prototype fidelity*) – nusako kiek realistiškas yra prototipas, kuo didesnis sudėtingumo (detalizavimo) lygis, tuo arčiau galutinio produkto yra pats prototipas, aukščiausio sudėtingumo lygio prototipo paprastai vartotojai neatskiria nuo tikro produkto.

Saityno programa (angl. *Web application*) – tai tokia programa, kuri yra įkelta nutolusiame serveryje ir kurią galima pasiekti internetu, dažniausiai per naršyklę.

Programų sąsaja (angl. *Application program interface*) – duomenų, funkcijų ir kitų reikalingų įrankių rinkinys, leidžiantis kurti tokias programas, kurios galėtų nesunkiai pasinaudoti suteikiamais resursais.

MVC (angl. *Model-View-Controller*) – programinės įrangos architektūrinis išskaidymo būdas, kuomet kodas išskaidomas į dalis, skirtas tik duomenų modeliams aprašyti, logikai aprašyti bei atvaizdavimo langams atskirai aprašyti.

Priklausomybių „injekcija“ (angl. *Dependency Injection*) – programavimo technika, leidžianti gauti viename objekte kito objekto savybes ir funkcijas, nuo kurių pirmasis objektas yra priklausomas.

Aprašomas tvirtai (angl. *Strongly typed*) – programavimo kalbos apibūdinimo būdas, kuomet turima omenyje, kad programavimo kalboje yra griežtai draudžiama atlikti tam tikrus veiksmus, pavyzdžiui, programos vykdymo metu pakeisti kintamojo tipą.

Aprašomas laisvai (angl. *Loosely typed*) – programavimo kalbos apibūdinimo būdas, kuomet turima omenyje, kad programavimo kalboje kintamųjų tipai neprivalo būti griežtai nurodomi ir žinomi visą programos vykdymosi laiką.

Įvadas

Problemos aktualumas

Kuriamas projektas turėtų palengvinti vartotojo sąsajos projektuotojo darbą supaprastinant idėjų generavimo ir pradinio projekto apipavidalinimo procesą. Toks projektas padėtų įmonėms sutaupyti laiko, kuris yra skiriamas projekto planavimo fazėje, nes idėjos būtų išgrynintos per trumpesnę laiko tarpą, o rezultatai būtų perteikiami daug labiau išdirbta forma, kuri būtų aiškesnė supratimui tiek įmonės klientams, tiek grafinio dizaino kūrėjams.

Šiuo metu tokia planavimo fazės dalis dažnai yra praleidžiama ir bandoma iškart piešti vartotojo sąsajos išdėstymą arba jau ir patį dizainą. Tačiau dažnai tai tenka perdarinėti dėl to kad yra nepataikoma į kliento norus dėl per silpno ar per ilgai užtrunkančio temos išnagrinėjimo ir dėl to yra švaistomas visų laikas ir finansiniai ištekliai. Kai kuriais atvejais vartotojo sąsajos projektuotojai ar dizaineriai piešia eskizus, kurie būna popierinio ir / arba skaitmeninio formato. Ranka piešiami popieriniai eskizai padeda greitai generuoti idėjas ir jas aptarti su projektų užsakovais. Tačiau tokie piešti eskizai nesuteikia pilno galimo vaizdo, kurį galima pamatyti tik iš skaitmeninio eskizo arba jau šiek tiek labiau apdirbto vidutinio-sudėtingumo prototipo (angl. *mid-fid prototype*). Jeigu norima pasiekti efektyvumą, kurį galima gauti iš ranka pieštų eskizų, bei išbaigtumo pilnumą, kurį galima pamatyti iš skaitmeninių eskizų, tai ranka nupieštus ir atrinktus eskizus dažniausiai kam nors tenka perpiešti kompiuteryje, kas gali užtrukti nuo pusvalandžio iki kelių valandų vienam projektuojamos programinės įrangos langui. Kuriamas projektas apjungtų šias dvi dalis į vieną, o tai įvyktų tokiu principu: vartotojo sąsajos projektuotojas nupiešia norimus eskizus; juos nufotografuoja arba nuskenuoja; įkelia į sistemą; sistema nuskaito eskizą, išskiria logines dalis, įsimena elementus ir pozicijas; sistema iš turimų duomenų nupiešia skaitmeninį vidutinio sudėtingumo prototipą arba kitaip vadinamą skaitmeninį eskizą. Tokiu darbo procesu nebereikėtų laukti kol kas nors nupieštų iš turimų eskizų skaitmeninius variantus ir rezultatus būtų galima peržiūrėti net ir per tą patį susitikimą su užsakovu per kurį yra aptarinėjami eskizai.

Darbo tikslas

Palengvinti eskizuose nupieštų vartotojo sąsajų skaitmenizavimą sukuriant sprendimą, kuris padėtų automatizuoti dalį šiuo metu rankiniu būdu atliekamų užduočių, taip pagreitinant idėjų generavimo procesą bei pateikiant darbo rezultatus greičiau.

Uždaviniai

1. Išanalizuoti panašios programinės įrangos (prototipų) veikimą bei aktualiausius susijusius kitus sprendimus.
2. Sudaryti kuriamos programinės įrangos reikalavimų specifikaciją, atsižvelgiant į analizės rezultatus ir projekto tikslą.
3. Realizuoti suprojektuotą sistemą.
4. Ištestuoti sistemą.
5. Atlikti eksperimentinius tyrimus sprendimo įvertinimui.

Darbo struktūra

Skyriuje „Analitinė dalis“ yra pateikiama literatūros ir technologijų analizė.

Skyriuje „Projektinė dalis“ yra pateikiami esminiai architektūriniai sukurtos programinės įrangos aspektai.

Skyriuje „Tyrimo dalis“ yra analizuojama sukurtos programinės įrangos kokybė, tyrinėjamos kokybės tobulinimo galimybės ir realizuojami tobulinimai.

Skyriuje „Eksperimentinė dalis“ yra atliekamas sukurtos programinės įrangos eksperimentinis tyrimas.

1. Analitinė dalis

1.1. Įvadas

Vartotojo sąsajos projektavimas yra svarbi programinės įrangos kūrimo dalis, nes tinkamai įgyvendinus šią dalį galima ne tik sutaupyti laiko kuriant pačią programinę įrangą dėl didesnio aiškumo, bet tai taip pat padeda geriau išsianalizuoti vartotojų poreikius bei parengti geresnę programinės įrangos reikalavimų specifikaciją. Šiuo metu yra įvairių metodikų kaip ir per ką galima projektuoti ir prototipuoti vartotojo sąsają. Kiekviena metodika turi savų privalumų bei trūkumų. Šio darbo tikslas sujungti gerąsias dviejų prototipavimo praktikų savybes.

Vartotojo sąsajai idėjas greičiausiai galima generuoti piešiant popierinius eskizus, tačiau daugiau aiškumo suteikia kompiuterinis prototipas. Tiek popierinis, tiek kompiuterinis variantas leidžia surasti labai panašų kiekį kritinių klaidų, tačiau kompiuterinis variantas paprastai jį testuojantiems vartotojams yra aiškesnis, todėl jie būna kritiškesni ir sugeba pastebėti daugiau, tačiau nedidelių galimų problemų. Šis darbas turėtų sujungti šias dvi dalis, leidžiant efektyviai generuoti idėjas, rinkti reikalavimus pasinaudojant popierinių eskizų teikiama privalumais bei aiškumu, gaunamu iš kompiuterinių prototipų ir suteikti galimybę greičiau bei patogiau testuoti sugalvotas vartotojo sąsajos idėjas.

Šio darbo sritis apims vartotojo sąsajų prototipavimo teoriją, vaizdų apdorojimą, eskizuose esančių simbolių atpažinimą, jų pozicijų nuskaitymą bei galutinio kompiuterinio prototipo generavimą.

1.2. Srities apžvalga

Šiame skyriuje bus apžvelgti vartotojo sąsajų prototipavimo būdai ir etapai, bus išskirti panašumai ir skirtumai tarp skirtingų prototipavimo būdų bei apibendrinta, iš ko susidaro eskizas arba prototipas. Taip pat bus panagrinėti įvairūs aspektai, susiję su vaizdų atpažinimu, tokie kaip pirminis vaizdų apdorojimas, figūrų ar simbolių nuskaitymas iš eskizo, figūrų padėčių nustatymas iš eskizo bei pačių diagramų nuskaitymas. Iš galutinio nuskaityto eskizo galima sugeneruoti galutinę vartotojo sąsają arba bent jau jos prototipą, todėl tai irgi bus apžvelgiama.

1.2.1. Vartotojo sąsajos prototipavimas

Norint išsiaiškinti potencialių programinės įrangos naudotojų norus bei iš to sudaryti tinkamus reikalavimus kuriamai programinei įrangai, galima dalį projekto projektavimo laiko skirti vartotojo sąsajos projektavimui. Vartotojo sąsaja gali būti prototipuojama įvairiais būdais, formatais, taikant skirtingas metodikas ir kt. Šiame poskyryje bus vartotojo sąsajos prototipavimo apžvalga.

Prototipų formatai

Vartotojo sąsają galima projektuoti ant popieriaus (angl. *paper-based prototype*), piešiant per kompiuterį (angl. *computer-based prototype*) arba įgyvendinant veikiančią vartotojo sąsajos prototipą įrenginyje, kuriam yra kuriamas galutinis produktas (angl. *fully functional prototype*) [1, 2, 3]. Paveiksliuke **1.1 pav.** matyti kaip atrodo kiekvienas iš šių formatų.



1.1 pav. Popierinio, kompiuterinio ir veikiančio prototipų palyginimas [2]

Vartotojo sąsajos prototipavimas ant popieriaus leidžia greičiau ir laisviau apžvelgti įvairias vartotojo sąsajos idėjas [2, 4]. Taip pat, į tokį prototipavimo būdą galima įtraukti daugiau mažesnių sugebėjimų žmonių, kadangi sąsajos piešimui ant popieriaus nereikia daug sugebėjimų [4, 5]. Prototipuodami sąsają ant popieriaus bei ją paskui peržvelgdami, vartotojai sugeba surasti tiek pat kritinių klaidų, kiek jų surastų ir testuojant visiškai baigtą prototipą [6]. Testuodami tokį popierinį prototipą visi daug lanksčiau žiūri į pakeitimus, kadangi ant popieriaus nupiešti naują eskizą nekainuoja daug laiko ar kitų resursų sąnaudų [5, 7, 8]. Tačiau popieriniai eskizai yra nelankstūs tuo aspektu, kad norint kažką pakeisti, dažniausiai tenka perpiešti visą eskizą bei nupiešus daug eskizų, kai kurie gali pasimesti, juos gali būti sudėtinga tvarkingai susidėti [1, 3, 9].

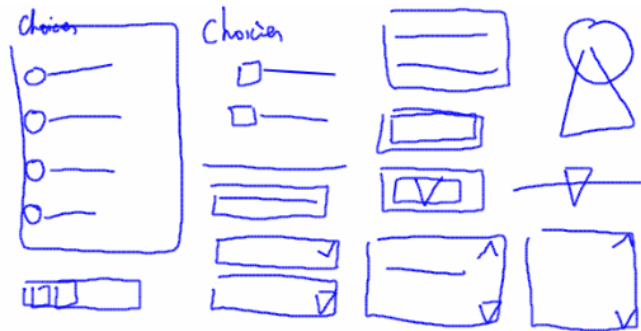
Kompiuterinius eskizus nupiešti užtrunka žymiai daugiau laiko nei popierinius eskizus, tačiau jie yra aiškesni ji testuojantiems naudotojams, bet tai nereikia, kad kompiuteriniai eskizai leidžia surasti daugiau kritinių klaidų, o tai gali skambėti kiek kontraversiškai [1, 7]. Kompiuterinio prototipo testavimo metu vartotojai patys supranta, kur ir ką gali spausti bei prototipą sugeba testuoti patys, priešingai nei popierinio prototipo testavimo metu, kai turi būti papildomas prižiūrėtojas, kuris ir valdytų testavimo procesą, atsižvelgdamas į vartotojo paspaudimus ant popieriaus lapo [5, 10].

Pilnai išbaigtas prototipas yra pritaikytas įrenginiui kuriam yra kuriamas galutinis produktas [2]. Toks prototipas leidžia suprasti kaip kuriama programinė įranga atrodys ir veiks būtent toje aplinkoje kuriai ir yra skirtas. Tokio tipo prototipai yra patys interaktyviausi ir aiškiausi, tačiau tokio tipo prototipą parengti užtrunka daugiausiai laiko, kadangi jis dažniausiai bus kuriamas jau kodu [5], pavyzdžiui, HTML / CSS, ar pasinaudojant tam tikros programavimo kalbos biblioteka, skirta vartotojo sąsajos kūrimui.

Prototipavimo būdai (sudėtingumo / detalizavimo lygiai)

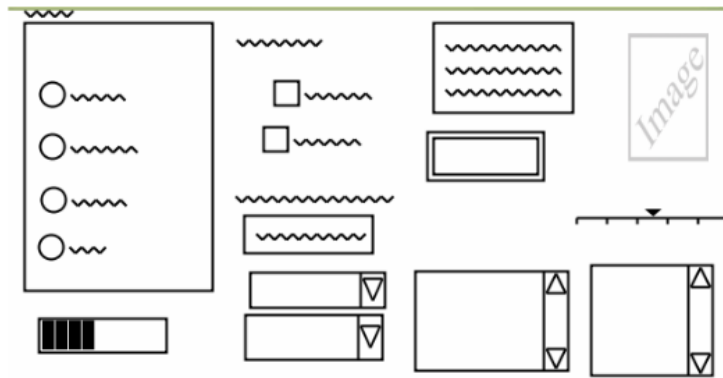
Vartotojo sąsajos prototipas gali būti įvairaus sudėtingumo lygio. Tiek popieriniai, tiek kompiuteriniai prototipai gali būti mažo (angl. *low-fid*), vidutinio (angl. *medium-fid*) arba didelio (angl. *high-fid*) sudėtingumo lygio [1, 3, 11, 12].

Mažo sudėtingumo lygio prototipas matomas 1.2 pav.



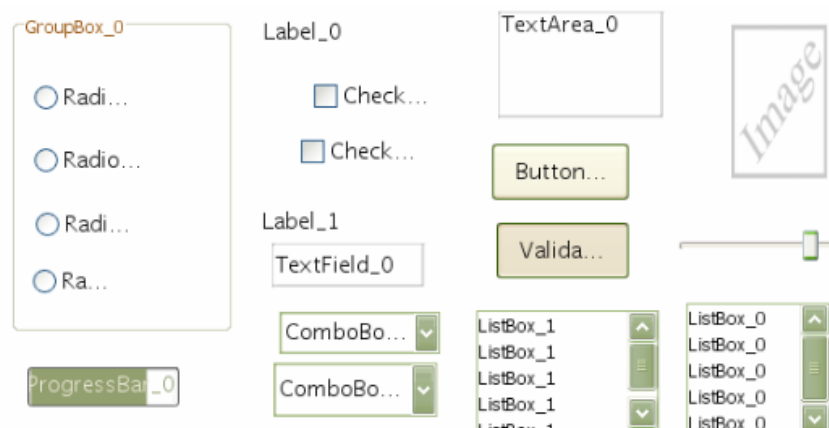
1.2 pav. Mažo sudėtingumo lygio prototipas [12]

Vidutinio sudėtingumo lygio kompiuterinis prototipas matomas 1.3 pav.



1.3 pav. Vidutinio sudėtingumo lygio prototipas [12]

Didelio sudėtingumo lygio kompiuterinis prototipas matomas 1.4 pav.



1.4 pav. Didelio sudėtingumo lygio prototipas [12]

Mažesnio sudėtingumo prototipai užtrunka žymiai mažiau laiko sąnaudų, kad tokius parengti, tačiau jie nėra tokie aiškūs kaip didesnio sudėtingumo prototipai [1, 11, 12]. Tačiau tai nereiškia, kad

mažesnio sudėtingumo lygio prototipai yra niekuo nenaudingi. Net ir mažo sudėtingumo lygio prototipai leidžia surasti daug potencialių problemų, kurias yra lengva ištaisyti prototipe bei kurias daug sudėtingiau ištaisyti jau sukurtoje programinėje įrangoje [6].

Mažo sudėtingumo lygio prototipai dažniausiai yra pilkų spalvų, nėra naudojami jokie sudėtingi grafiniai elementai, tokie kaip šešėliai, tokiuose prototipuose nebūtinai naudojamas galutinis tekstas, vietoj kurio yra piešiami paprasčiausios linijos bei naudojami kiti grubiai nupiešti elementai [11]. Didelio sudėtingumo prototipai dažniausiai atrodo kaip galutinė programinė įranga, dėl to labai lengva įsivaizduoti ko galima tikėtis sulaukus galutinio produkto [11]. Didžiausi skirtumai tarp šių prototipų yra aiškumas jį testuojantiems naudotojams, o svarbiausia laiko sąnaudos, kurios yra investuojamos į tokių prototipų sukūrimą. Mažo sudėtingumo lygio prototipą sukurti gali užtrukti žymiai mažiau laiko nei didelio sudėtingumo lygio prototipą [1, 11]. Vidutinio sudėtingumo prototipas yra tarpinis variantas tarp mažo ir didelio sudėtingumo prototipo. Tokiame prototipe dar nebūtinai bus naudojamos spalvos ar kiti galutinio projekto grafiniai elementai, tačiau jame jau bus nupieštos tvarkingiau atrodančios figūros, bus aiškiau kur yra tekstas, o kur paprastos linijos, galbūt bus panaudoti pilkos spalvos atspalviai, kad atskirt logiškas dalis vieną nuo kitos.

1.2.2. Pirminis vaizdų apdorojimas

Norint pagerinti nufotografuotus arba nuskenuotus ant popieriaus nupieštus prototipus (popierinius eskizus), prieš bandant atpažinti simbolius ir diagramas, reikėtų tuos vaizdus apdoroti, kad būtų pakelta piešinio kokybė ir dirbtiniam intelektui būtų suteiktos geresnės sąlygos tų vaizdų atpažinimui. Vaizduose galima apdoroti įvairius dalykus, tokius kaip:

- Sąsiuvinio linijų ar langelių šalinimas [13].
- Simbolių ar figūrų eskize linijų storio keitimas [13, 14, 15, 16].
- Iškreipimo (angl. *skew*) arba perspektyvos (angl. *perspective*) nustatymas ir sutvarkymas [13, 14, 17].
- Triukšmo šalinimas [14, 16, 18, 19].
- Linijų ištiesinimas ir pagerinimas (angl. *beautification*) [20, 21, 22].
- Linijų kokybės kėlimas pašalinant pašalinį žymėjimą iš jų, pačias linijas sujungiant tarpusavyje, jei yra dėl kažko perkirstos [14, 16, 18].
- Linijų savybių išskyrimas (angl. *feature extraction*) [14, 15, 19].
- Dydžių normalizacija [14, 21].
- Geometrinės transformacijos [15, 19].
- Linijų kabliukų šalinimas (angl. *elimination of hooks*), kurie dažniausiai atsiranda linijos piešimo pradžioje arba pabaigoje [21].
- Visos nuotraukos pasukimas [17].
- Pavertimas į nespaltotą vaizdą [19].
- Kampų suaštrinimas [19].

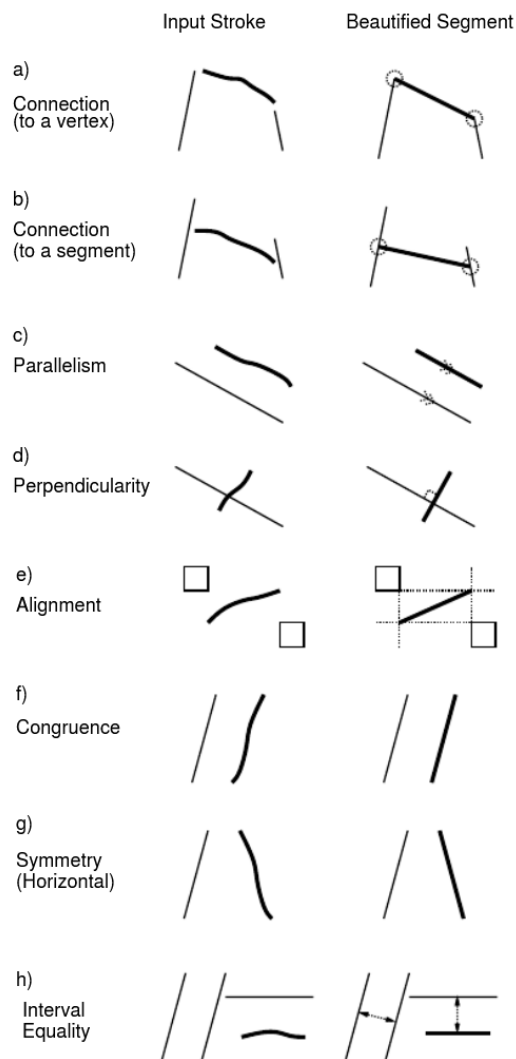
Vaizdų apdorojimo technikos

Sąsiuvinio linijų nustatymas ir šalinimas matomas **1.5 pav.** Ši metodika turėtų būti naudojama tuomet kai papildomas triukšmas, šiuo atveju sąsiuvinio linijos, gali trukdyti tolimesniam vaizdo apdorojimui [13].

Original Word	Following Underline Removal

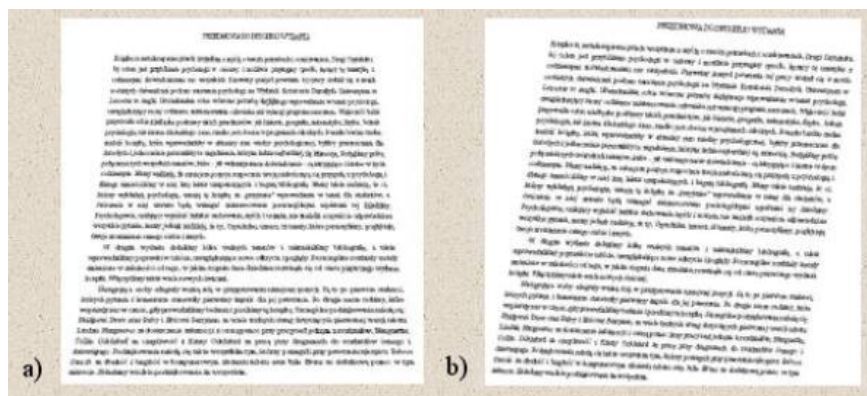
1.5 pav. Linijų šalinimas apdorojant vaizdą [13]

Linijų tiesinimas ir gražinimas (angl. *beautification*) matomas 1.6 pav.. Šią metodiką naudoti pravartu kai reikia šiek tiek aptvarkyti netvarkingai nupieštas geometrines figūras. Taikant gražinimą (angl. *beautification*) ant apdorojamo vaizdo, galima gauti šiek tiek tvarkingiau perpieštą vaizdą [23].



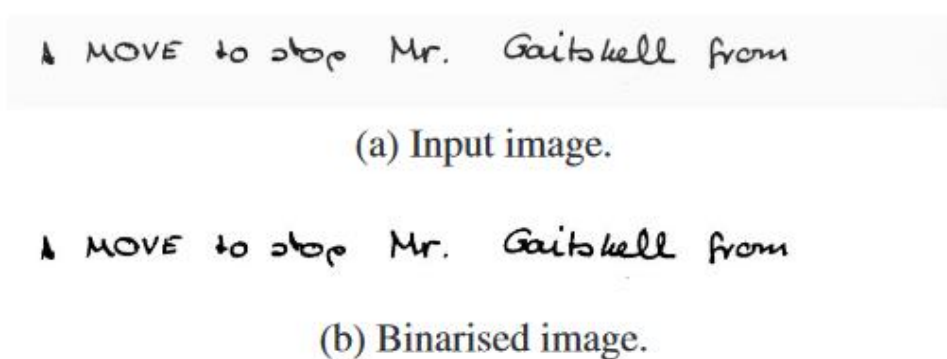
1.6 pav. Linijų tiesinimas ir gražinimas apdorojant vaizdą [23]

Jeigu bus apdorojamas vaizdas, kuris bus nufotografuotas, o ne nuskenuotas, tikėtina, kad vaizdo perspektyva bus iškreipta, kaip tai matoma **1.7 pav.** [17]. Dėl iškreiptos vaizdo perspektyvos gali pasidaryti sunku arba neįmanoma iš eskizo nuskaityti atvaizduojamo prototipo vaizdo ir jame esančio žymėjimo.



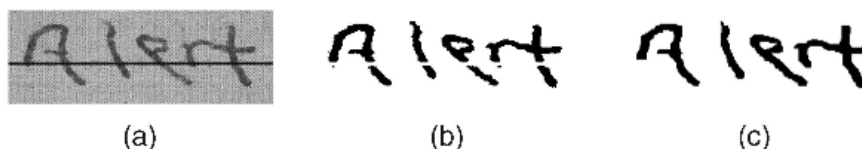
1.7 pav. Atvaizdas su netvarkinga perspektyva [17]

Norint palengvinti nupieštų simbolių ar figūrų atpažinimą iš eskizo, galima papildomai apdoroti pačias tokių figūrų linijas, pritaikant linijų storio keitimo algoritmus, nutrūkstančių linijų apjungimus į bendras linijas, triukšmo šalinimą ir kt.. Vienas iš pavyzdžių, kuriame matomas pradinis vaizdas ir galutinis rezultatas matomas **1.8 pav.**



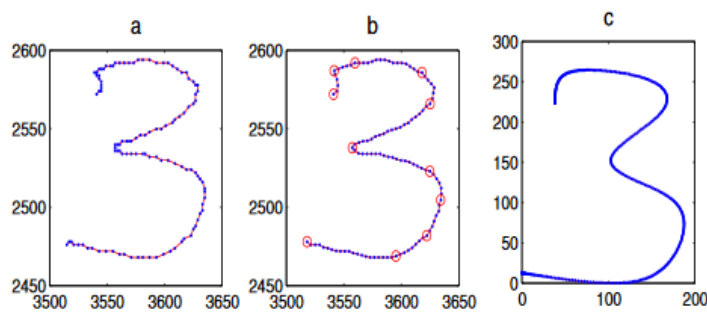
1.8 pav. Linijų aptvarkymas vaizde [16]

Taip pat, pavyzdys kai yra šalinamas triukšmas, papildomos sąsiuvinio linijos, nepriklausančios pagrindiniam vaizdui, aptvarkymas bei apjungimas matomas **1.9 pav.**



1.9 pav. Linijų aptvarkymas vaizde [18]

Linijų aptvarkymo vaizde pavyzdys kai yra šalinami linijų kampai ir pritaikoma normalizacija matoma **1.10 pav.**



1.10 pav. (a) pradinis vaizdas, (b) linijų kampų šalinimas, (c) linijų normalizacija [21]

1.2.3. Iššūkiai eskizų nuskaityme, modelio atpažinime

Pirmasis iššūkis, kodėl gali kilti problemų, tai prasmingų figūrų, simbolių suradimas eskizuose [24]. Žmonės piešia eskizus šiek tiek skirtingai, todėl vaizdų atpažinimo algoritmai turėtų būti pakankamai lankstūs, kad sugebėtų atpažinti tą pačią prasmę turinčius simbolius ar figūras, net jeigu jos kažkiek skiriasi, tačiau turėtų būti pakankamai apibrėžtas, kad nenuskaitytų kažko kas yra ne į temą [24].

Taip pat, nuskaitymą gali būti sudėtinga atlikti dėl dviprasmybių, pasitaikančių eskizuose, nes kartais ta pati linija gali reikšti skirtingus dalykus: ar tai yra paprasta linija, reiškianti elementų atskyrimą; ar tai yra linija, žyminti paragrafą; ar ta linija yra dalis didesnės figūros; ar kas nors kita [24]. Kad būtų galima išspręsti šią problemą programa kiekvienu atveju turi surasti ar nupiešta linija ar kitas simbolis gali turėti daugiau nei vieną reikšmę ir jeigu turi, tuomet parinkti artimiausią variantą [24]. Tačiau jeigu programa per stipriai bandys apibrėžti rezultatus, tai gali erzinti vartotojus, nes gali daugėti atvejų kai nustatomos netinkamos figūros [24].

Dar viena problema bandant apjungti natūralų žmogaus piešimą ir kompiuterio darbą, tai kaip tie rezultatai yra apjungiami tarpusavyje [24]. Kai kurios programos palieka vartotojo eskizą tokį koks yra ir kompiuterines interpretacijas pateikia atskirame lange, tačiau tai verčia vartotoją nukreipti dėmesį nuo eskizo ir ieškoti ar kiekviena linija buvo interpretuota tinkamai [24]. Alternatyvus metodas būtų interpretuoti vartotojo linijas tuo metu kai jis jas piešia, tuomet vartotojas iškart matytų ar kompiuteris tinkamai suprato tai ką jis norėjo nupiešti, tačiau jeigu sistema linijas interpretuoja per anksti ar per agresyviai, tai gali erzinti vartotoją [24].

Sprendžiant neteisingų interpretacijų klaidas yra dar vienas iššūkis su kuriuo teks susidurti. Bet kuri nuskaitymo programa anksčiau ar vėliau padarys klaidų interpretuojant vartotojo eskizą, dėl kylančių dviprasmybių eskizuose ar dėl blogai apmokyto atpažinimo variklio [24]. Optimaliausiu atveju vartotojas turėtų matyti klaidas iškart kai tik jos įvyksta ir turėtų galėti jas patogiai ištaisyti nepametant minties ką darė prieš tai [24].

Tikriausiai didžiausia problema iš visų tai variklio sukūrimas, kuris sugebėtų tinkamai atpažinti objektus eskize, tačiau kartu ir būtų pakankamai lankstus, kad jo darbo rezultatus būtų galima pakeisti [24]. Praktikoje neįmanoma sukurti variklio, kuris nedarytų klaidų, tačiau galima siekti sukurti tokį variklį, kuris padarytų kaip įmanoma mažiau klaidų. Šią problemą galima išskaidyti į dvi konceptualias dalis: atpažįstant visas įmanomas interpretacijas; bei nusprendžiant kuri interpretacija yra labiausiai tikėtina esamame kontekste [24].

1.2.4. Kompiuterinio matymo technikos (angl. *computer vision techniques*)

Norint tinkamai klasifikuoti eskizuose esančius elementus galima naudoti įvairias kompiuterinio matymo technikas:

- Triukšmo slopinimas (angl. *image denoising*) – kadangi naudojamos realaus pasaulio nuotraukos, o ne kompiuteriu piešti paveikslukai, tokiose nuotraukose dažnai būna triukšmo dėl kamerų jutiklių skirtumų ir tai gali sukelti problemų bandant atpažinti kraštus [25].
- Spalvos atpažinimas (angl. *color detection*) – spalvos atpažinimas ar, tiksliau, slenksčio tarp spalvų ir jų savybių atpažinimas, gali padėti aptinkant elementus eskize [25].
- Kraštinių aptikimas (angl. *edge detection*) – elementus eskize galima rasti aptinkant jų kraštines, o kraštinių aptikimo metodikų yra daug, bet kelios populiariausios yra „*The Sobel operator*“, „*The Canny operator*“, „*Richer Convolutional Features*“ [25, 26, 27, 28].
- Segmentacija (angl. *segmentation*) – eskizuose gali būti daug elementų, todėl juos suklasifikuoti turėtų padėti segmentacija, yra daug galimų segmentacijos algoritmų, tačiau populiariausi yra „*color based*“, „*structure based*“, „*trainable segmentation*“ [25].
- Teksto atpažinimas (angl. *text detection*) – norint atpažinti tekstą eskizuose, galimos įvairios technikos, pagrindinės grupės būtų „*Deep learning methods*“ (pavyzdžiui, CNN paremti metodai) ir „*Feature based methods*“ [25, 27].

1.2.5. Mašininio mokymosi technikos

Mašininio mokymosi technikos padeda kompiuteriams išmokti duomenis, nerašant konkrečių algoritmų kaip tuos duomenis išmokti. Eskizų nuskaitymo užduotis yra viena iš tokių užduočių, kuriai labai pasitarnautų mašininis mokymasis [25].

Dirbtiniai neuroniniai tinklai (angl. *Artificial Neural Networks (ANNs)*)

Dirbtinis neuroninis tinklas yra supaprastintas skaičiavimų modelis (angl. *computational model*), kuris sukurtas žmogaus smegenyse esančių neuroninių tinklų pagrindu [25]. Neuronas skaičiuoja įvesčių svorių vidurkius, kurie yra perduodami netiesinei funkcijai kaip aktyvacijos funkcija, tuomet rezultatai perduodami visoms išvestims jas pakoreguojant [25]. Į priekį einantis (angl. *feedforward*) ANN tai toks tinklas, kur duomenys eina tik į vieną pusę, jame nėra ciklų, tuo tarpu pasikartojantis (angl. *recurrent*) ANN turi ciklus [25].

Daugiasluksniai perceptroniniai tinklai (angl. *Multilayer Perceptron Networks (MLPs)*)

Daugiasluksnis perceptroninis tinklas yra į priekį einančio dirbtinio neuroninio tinklo (angl. *feedforward ANN*) vienas iš porūšių [25]. MLPs susideda iš kelių daug neuronų skirtinguose sluoksniuose, kiekvienas neuronas turi jungtis su neuronais iš ankstesnio sluoksnio, o tai vadinama pilnai sujungtu (angl. *fully connected*), taip pat, kiekviena jungtis turi svorį [25]. Yra šie sluoksnių tipai: [25]

- Įvesčių sluoksnis (angl. *input layer*) – pirmasis tinklo sluoksnis, neatliekantis jokių skaičiavimų, tiesiog perduoda informaciją sekantiems sluoksniams.
- Paslėpti sluoksniai (angl. *hidden layers*) – nulis arba daugiau sluoksnių, kurie yra iškart už įvesčių sluoksnio, kuriuose yra atliekami skaičiavimai ir viskas perduodama į išvesčių (rezultatų) sluoksnį.

- Išvesčių (rezultatų) sluoksnis (angl. *output layer*) – paskutinis tinklo sluoksnis, atliekantis skaičiavimus, perduodantis galutinius rezultatus iš tinklo.

Kai MLP yra apmokomas, įvesčių ir išvesčių duomenys yra perduodami tinklui, ir algoritmas (atgalinis užpildymas (angl. *backpropagation*)) atnaujina visus svoris tarp neuronų [29]. Kuo sudėtingesnę informaciją turi išmokti tinklas, tuo daugiau duomenų reikia ir tuo sudėtingesnis tinklas darosi [25]. Po apmokymo tinklui galima perduoti norimus duomenis ir tinklas grąžins kažkokį atsakymą.

Sluoksnių plotis (angl. *width*) yra neuronų tame sluoksnyje skaičius, o kuo platesnis sluoksnis – tuo geriau gali įsiminti informaciją [29]. Tinklo gylis (angl. *depth*) yra paslėptų sluoksnių skaičius, o kuo gilesnis tinklas – tuo skirtingesnes ypatybes iš skirtingų abstrakcijos lygių tinklas gali išmokti [29].

Generalizacija (angl. *generalisation*) nurodo tinklo veikimą kai jis gauna visiškai naujų duomenų. Tinklo persimokymas (angl. *overfitting*) kyla tuomet kai sukurtame modelyje ryšys tarp įvesties duomenų ir išvesties duomenų yra per daug griežtas, o tai lemia prastą generalizaciją [29]. Sudėtingesni tinklai (platesni ir gilesni) labiau linkę į persimokymą [29].

Yra daug kitų mašininio mokymosi technikų, ką būtų galima panaudoti vietoj MLPs, pavyzdžiui, palaikančias vektorių mašinas (angl. *support vector machines* (SVM)), ar atsitiktinius miškus (angl. *random forests*) [25, 30, 31].

Gilasis mokymasis (angl. *Deep Learning*)

Gilasis mokymasis yra mašininio mokymosi kryptis, kur naudojami gilūs neuroniniai tinklai, kurie sudaryti iš daug paslėptų sluoksnių [25]. Gilasis mokymasis parodė gerus rezultatus įvairiose srityse, pranokstant įvairius klasikinius metodus [25].

Konvoliuciniai neuroniniai tinklai (angl. *Convolutional Neural Networks* (CNN))

Konvoliucinis neuroninis tinklas tai specifinė neuroninio tinklo rūšis, kuri lengvai imituoja gyvūnų regimojoje žievėje randamų neuronų jungties modelį [25]. CNN yra ypač naudingi atliekant vaizdų užduotis, tačiau gali būti pritaikyti kitoms užduotims, pavyzdžiui, natūraliai kalbai apdoroti [25].

CNN yra ypač naudingi, palyginti su MLP, atliekant vaizdo apdorojimo užduotis, nes MLP gali turėti problemų su nuotraukų matmenimis dėl visiško neuronų susietumo, todėl prastai veikia su didelės raiškos vaizdais [25]. Be to, MLP neatsižvelgia į duomenų erdvinę struktūrą, todėl taškai, esantys priešingose vaizdo pusėse, būtų traktuojami taip pat, kaip ir taškai, esantys arti vienas kito [25].

CNN sudaryti iš įvesties, paslėptų ir išvesties sluoksnių. Paslėpti sluoksniai gali būti šie: [25]

- Konvoliuciniai sluoksniai (angl. *convolutional layers*) – kurie atlieka konvoliucinius veiksmus ant įvesties, perduodant rezultatus į kitą sluoksnį.
- Kaupimo sluoksniai (angl. *pooling layers*) – kurie sujungia rezultatus iš kelių neuronų į vieną neuroną kitame sluoksnyje, tai padeda mažinti tinklo dydį, kai siekiama sumažinti parametru ir skaičiavimų kiekį, tai padeda kontroliuoti persimokymą.
- Visiškai sujungti sluoksniai (angl. *fully connected layers*) – sujungia kiekvieną neuroną viename sluoksnyje su kiekvienu neuronu kitame sluoksnyje.

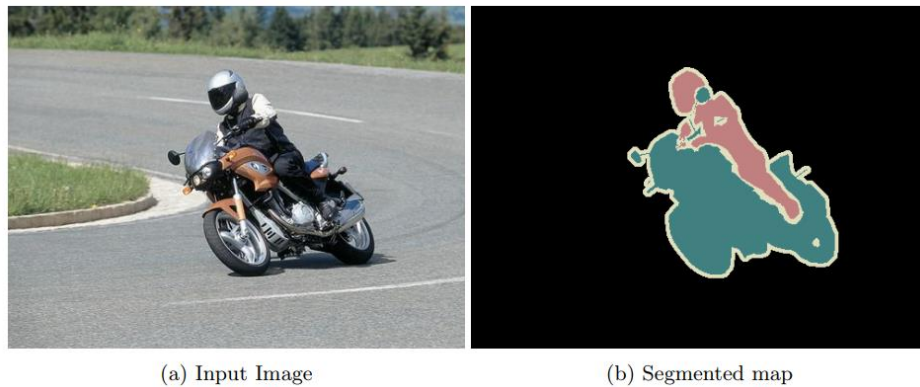
CNN turi šias pagrindines skiriamąsias savybes: [25]

- 3D neuronų tūriai (angl. *3D volumes of neurons*);
- vietinis ryšys (angl. *local connectivity*);
- svorių pasidalinimas (angl. *shared weights*).

Šios savybės padeda CNN geriau veikti su vaizdo užduotimis (angl. *vision problems*). Pavyzdžiui, svorių pasidalinimas leidžia sumažinti naudojamų parametrų kiekį, tai sumažina naudojamos atminties kiekį.

Semantinis segmentavimas (angl. *Semantic Segmentation*)

Vaizdo segmentavimas yra įprasta vaizdo apdorojimo užduotis, tikslas yra supaprastinti ir (arba) pakeisti vaizdą kažkuo prasmingesniu. Tai reiškia, kad kiekvienas pikselis pažymimas taip, kad taškai, turintys bendras charakteristikas, būtų pažymėti kartu (**1.11 pav.**). Yra keletas metodų, tokių kaip grupavimo metodai (angl. *clustering methods*), „*Texton Forests*“, kraštų aptikimas (angl. *edge detection*), apmokomi segmentavimo tinklai (angl. *trainable segmentation networks*) ar kita [25].



1.11 pav. Semantinio segmentavimo pavyzdys [32]

1.2.6. Kodo generavimas

Procesas kurio metu programos kūrėjas iš nupiešto dizaino sukuria programos grafinę vartotojo sąsają yra daug laiko reikalaujanti užduotis, kuri nukreipia programuotojų dėmesį nuo svarbesnių užduočių, taip pat, grafinių vartotojų sąsajų kodas yra labai specifiškas ir apibrėžtas, todėl tai greitai tampa varginančiu ir monotonišku darbu [33].

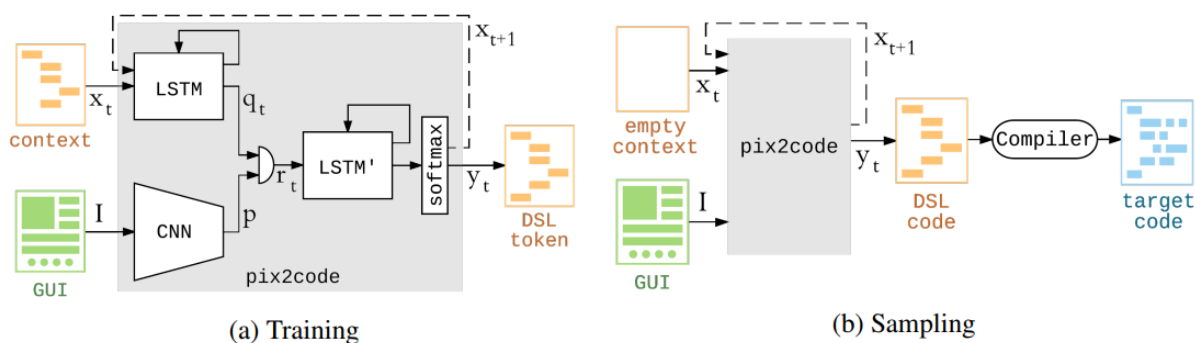
DeepCoder sprendimas padeda sugeneruoti kompiuterines programas pasinaudojant statistinėmis prognozėmis [34]. *Terpet* sprendime kodo generavimo galimybės buvo pasiektos dirbtiniam intelektui išmokstant ryšius tarp įvesties ir išvesties pavyzdžių per diferencijuojamus interpretorius [35]. Taip pat, keli autoriai yra aprašę programos kūrimo metodą iš natūralios kalbos ir struktūruotos programos specifikacijos kaip įvesčių [36]. Tačiau reikia turėti omeny, kad beveik visi šie sprendimai kliaujasi domenų specifinėmis kalbomis (angl. *Domain Specific Languages (DSLs)*), kompiuterių kalbomis (pavyzdžiui, žymėjimo, programavimo, ar modeliavimo kalbomis), kurios yra sukurtos konkrečiai taikymo sričiai, o tai gali limituoti iki kokio sudėtingumo būtų galima išplėsti šiuos sprendimus [33].

Norint išnaudoti grafinę įvesties pobūdį, galima pasiskolinti metodų iš kompiuterinių matymo technikų literatūros, kur galima rasti daug sėkmingų rezultatų, kurie parodo, kad gilieji neuroniniai tinklai sugeba išmokyti latentinių kintamųjų, apibūdinančių vaizde esančius objektus ir jų ryšį su

atitinkamais kintamo ilgio tekstiniais aprašymais [33]. Visi šie metodai paremti dviem pagrindiniais komponentais:

- Konvoliuciniais neuroniniais tinklais (angl. *Convolutional Neural Network*) atliekančiais neprižiūrimą (angl. *unsupervised*) ypatybių išmokimą, susiejantį neapdorotus duomenis (angl. *raw input*) su jau išmoktais duomenimis.
- Rekurentiniais neuroniniais tinklais (angl. *Recurrent Neural Network*) atliekančiais kalbos modeliavimą iš tekstinio aprašymo, kuris susijęs su įvesties nuotrauka.

Sprendimo *pix2code* architektūra matoma **1.12 pav.**. Atliekant apmokymą (**1.12 pav.** (a)) grafinės vartotojo sąsajos paveikslėlis yra apdorojamas su CNN modeliu, konteksto kalbos modelis apdorojamas su LSTM sluoksniais, tuomet apdoroti duomenys yra apjungiami ir apdorojami su antra LSTM sluoksnių grupe bei galiausiai *softmax* sluoksnyje yra apdorojamas kiekvienas gautas rezultatas (angl. *token*). Naudojantis modeliu (**1.12 pav.** (b)) galima perduoti norimą įvesties kontekstą, dėl ko bus sugeneruotas kodas norimai aplinkai.



1.12 pav. *pix2code* modelio architektūra [33]

1.3. Apibendrinančios išvados

1. Analizuojant vartotojo sąsajų projektavimo ir vartotojo sąsajų testavimo tyrimus, buvo pastebėta, kad galvoti naujas idėjas yra geriausiai, kai žmogus nėra apribojamas su kompiuterio turimais apribojimais, todėl tokiu atveju idėjas geriausiai piešti ant popieriaus. Taip pat, buvo pastebėta, kad visas pagrindines problemas, esančias vartotojo sąsajoje, galima surasti tiek popieriniuose eskizuose, tiek skaitmeniniuose prototipuose, tačiau smulkesnes problemas galima rasti tik skaitmeniniuose prototipuose. Bet turint popierinį eskizą, jo pavertimas į skaitmeninį prototipą gali užtrukti nemažai laiko, kurį projektuotojas galėtų panaudoti kitų užduočių atlikimui. Todėl šiame darbe bus tyrinėjamos galimybės kaip apjungti abiejų projektavimo būdų gerąsias savybes ir padėti projektuotojui greičiau pasiekti norimų rezultatų.
2. Analizuojant literatūrą buvo pastebėta, kad yra įvairių idėjų ir metodų kaip spręsti eskizo elementų atpažinimo ir jo transformavimo į kitą formatą problemą, yra atlikti keli eksperimentiniai darbai. Tačiau dar nėra pilnai aišku ar juos galima pritaikyti sprendžiant realaus pasaulio problemas, todėl šiame darbe bus mėginama ištyrinėti ir patobulinti dabar egzistuojančius sprendimus bei atlikti eksperimentinius tyrimus su kuriais bus mėginama išsiaiškinti tokio sprendimo efektyvumą.

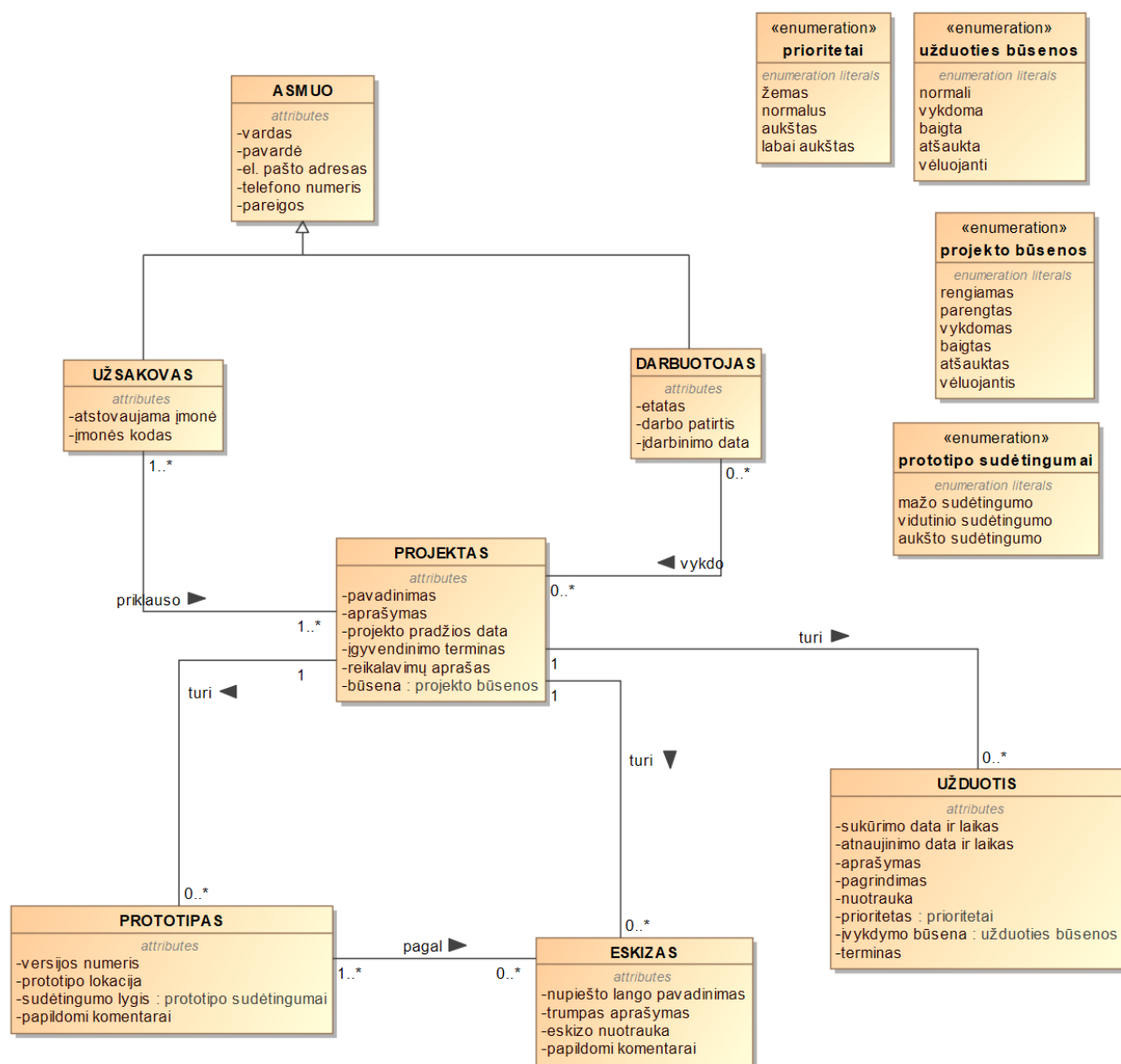
2. Projektinė dalis

Šiame skyriuje pateikta esminiai sukurtos programinės įrangos architektūriniai sprendimai.

2.1. Duomenų modelis ir jo elementų žodynas

2.1.1. Duomenų modelis

Sistemos duomenų modelį sudaro projektai, asmenys, kurie skirstomi į užsakovus ir darbuotojus, užduotys, prototipai, eskizai. Tokio duomenų modelio diagrama matoma 2.1 pav.



2.1 pav. Duomenų modelio diagrama

2.1.2. Duomenų žodynas (duomenų modelio specifikacija)

Pagrindinės duomenų modelio specifikacijos matomos nuo 2.1 lentelė iki 2.3 lentelė. Likusios duomenų modelio specifikacijos pateiktos priede „1 priedas. Duomenų modelio specifikacijos“.

2.1 lentelė. Duomenų modelio „Projektas“ klasės ir jos atributų aprašymas

Duomenų žodyno elemento pavadinimas	Turinys	Tipas
Projektas	Programinės įrangos kūrimo projektas. Užsakovo ar kelių užsakovų pageidavimu. Vykdomas įmonės darbuotojų.	Klasė
Pavadinimas	Kaip projektas bus vadinamas. Pavadinimas bus naudojamas projektą kuriančios įmonės viduje, tam kad eitų šį konkretų projektą atskirti nuo kitų projektų. Būtina nurodyti bet kokia tekstine forma.	Atributas
Aprašymas	Visa pagrindinė informacija apie projektą. Projekto kūrimo priežastys, tikslai, ką projektas palies, konkurentai, projekto apribojimai ir t. t. Turi būti nurodyti bent svarbiausia informacija be kurios negalėtų vykti darbas.	Atributas
Projekto pradžios data	Nuo kada projektas bus pradedamas įgyvendinti. Saugoma formatu YYYY-MM-DD. Nurodyti būtina.	Atributas
Įgyvendinimo terminas	Užsakovo nurodytas norimas arba būtinas projekto sukūrimo terminas. Saugoma formatu YYYY-MM-DD. Nurodyti nebūtina jei nėra jokio termino, kurio reikėtų laikytis.	Atributas
Reikalavimų aprašas	Funkcinių ir nefunkcinių reikalavimų sąrašas arba prisegtas dokumentas su šiais reikalavimais. Nurodyti būtina.	Atributas
Būsena	Kokia yra projekto įgyvendinimo būsena. Būsenai galima nurodyti tik iš anksto numatytą reikšmę. Galimos būsenos: rengiamas; parengtas; vykdomas; baigtas; atšauktas; vėluojantis. Projektas būtinai turi turėti kokią nors būseną.	Atributas

2.2 lentelė. Duomenų modelio „Prototipas“ klasės ir jos atributų aprašymas

Duomenų žodyno elemento pavadinimas	Turinys	Tipas
Prototipas	Sistemos projektuotojo darbo rezultatai, kuomet vartotojo sąsajos eskizus perpiešia į mažo arba vidutinio sudėtingumo lygio prototipą kompiuterio pagalba.	Klasė
Versijos numeris	Prototipo versijos numeris. Numeruojamas automatiškai sistemos, būtinas. Šis versijos numeris leis suprasti kuri prototipo versija yra peržiūrima.	Atributas
Prototipo lokacija	Lokacija kur galima rasti veikiančią prototipą. Nurodyti būtina.	Atributas
Sudėtingumo lygis	Koks yra prototipo sudėtingumo lygis. Būtina nurodyti vieną iš galimų reikšmių: mažo sudėtingumo; vidutinio sudėtingumo; aukšto sudėtingumo.	Atributas
Papildomi komentarai	Jeigu projektuotojas turi papildomų komentarų apie šią prototipo versiją, tai gali užrašyti. Saugoma paprasta tekstine forma. Neprivalomas.	Atributas

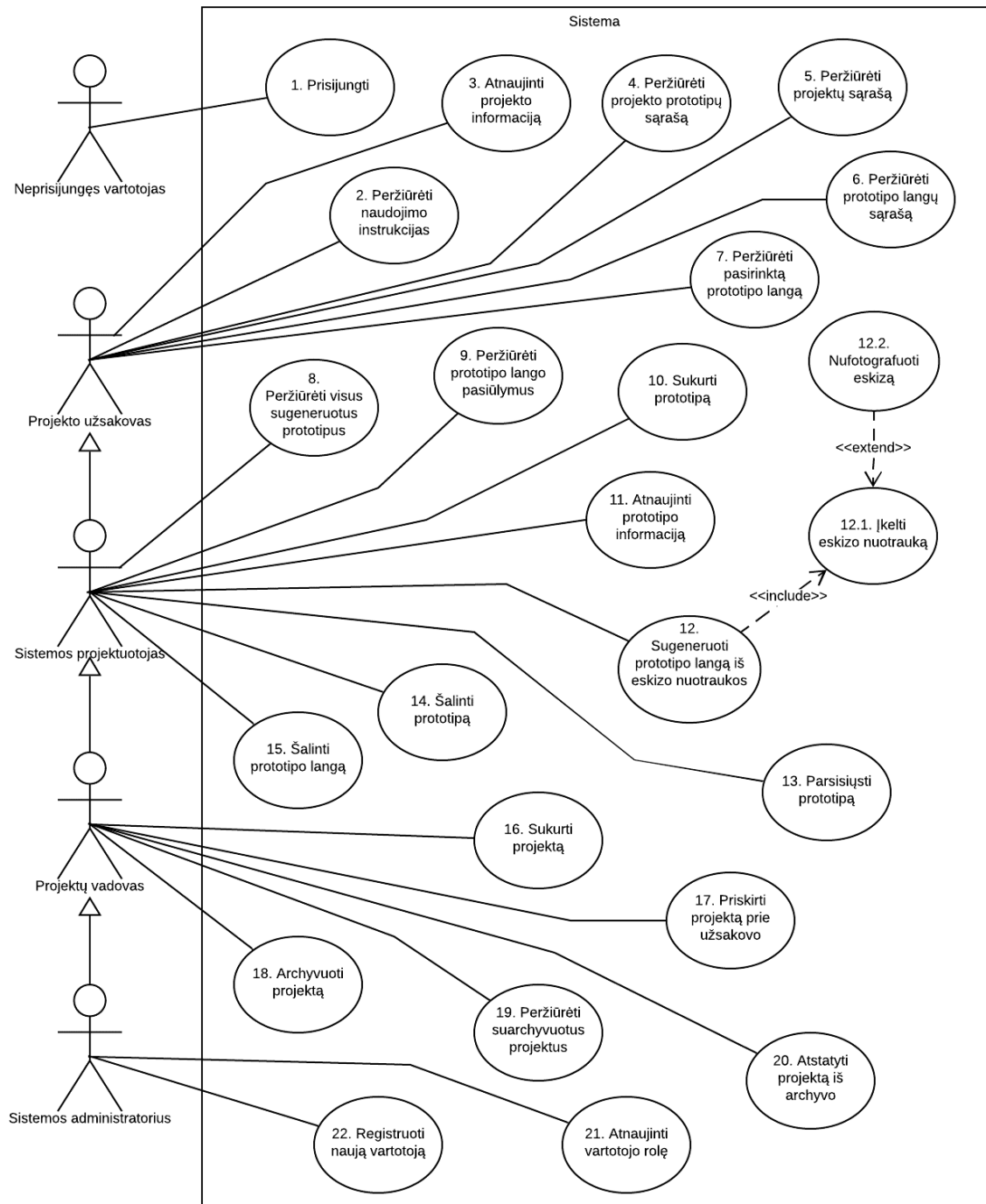
2.3 lentelė. Duomenų modelio „Eskizas“ klasės ir jos atributų aprašymas

Duomenų žodyno elemento pavadinimas	Turinys	Tipas
Eskizas	Sistemos projektuotojo pradiniai vartotojo sąsajos projektavimo darbo rezultatai.	Klasė
Nupiešto lango pavadinimas	Lango ar sistemos dalies, kurią pademonstruoja eskizas, pavadinimas. Nurodoma paprasta tekstine forma. Nurodyti būtina.	Atributas
Trumpas aprašymas	Galima pridėti papildomą aprašymą. Nebūtina.	Atributas

Eskizo nuotrauka	Nuskenuota arba nufotografuota eskizo nuotrauka. Leidžiami formatai: jpg, png, įkelti būtina.	Atributas
Papildomi komentarai	Jeigu yra papildomų komentarų, kuriuos projektuotojas nori pateikti. Nebūtina.	Atributas

2.2. Sistemos panaudos atvejų diagrama

Sistemos panaudos atvejų diagrama matoma **2.2 pav.** Sistemoje egzistuoja kelios vartotojų rolės. Atitinkamai pagal tai kokią rolę turi vartotojas, jis gali atlikti tai rolei leidžiamus veiksmus. Pagrindiniai sistemos panaudos atvejai yra: projektų, prototipų ir prototipų langų kūrimas; eskizo įkėlimas; eskizo nuskaitymas ir transformavimas į kitą norimą formatą.



2.2 pav. Panaudos atvejų diagrama

Pagrindinių panaudos atvejų specifikacijos aprašytos nuo **2.4 lentelė** iki **2.5 lentelė**. Likusių panaudos atvejų specifikacijos pateiktos priede „2 priedas. Panaudos atvejų specifikacijos“.

2.4 lentelė. Panaudos atvejo „Sugeneruoti prototipo langą iš eskizo nuotraukos“ specifikacija

Panaudojimo atvejis	PA-12. Sugeneruoti prototipo langą iš eskizo nuotraukos	
Aprašymas	Vartotojas su atitinkama role gali sugeneruoti prototipą (ar prototipo langą) iš turimo eskizo. Vartotojas gali pats įkelti nuskenuotą nuotrauką arba nufotografuoti per sistemą.	
Prieš-sąlyga	Vartotojas turi turėti projektuotojo, projektų vadovo arba sistemos administratoriaus rolę.	
Aktorius	Sistemos projektuotojas; projektų vadovas; sistemos administratorius	
Susiję panaudojimo atvejai	Išplečiantys PA (<<extend>> ryšys)	12.2. Nufotografuoti eskizą
	Apimami PA (<<include>> ryšys)	12.1. Įkelti eskizo nuotrauką
Pagrindinis scenarijus		
Po-sąlyga	Atvaizduojamas sugeneruoto prototipo langas.	

2.5 lentelė. Panaudos atvejo „Parsisiųsti prototipą“ specifikacija

Panaudojimo atvejis	PA-13. Parsisiųsti prototipą
Aprašymas	Vartotojas su atitinkama role gali parsisiųsti sugeneruotą prototipą.
Prieš-sąlyga	Sistemoje turi būti sukurtas bent vienas prototipas, kurį būtų galima parsisiųsti. Vartotojas turi turėti projektuotojo, projektų vadovo arba sistemos administratoriaus rolę.
Aktorius	Sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram actor Vartotojas Vartotojas->>Vartotojas: Prototipo peržiūra Vartotojas->>Sistema: Užklausa dėl prototipo parsisiuntimo Sistema-->>Vartotojas: Parodyti galimus formatus Vartotojas->>Vartotojas: Pasirinkti formatą Vartotojas->>Sistema: Pradėti siuntimą Sistema-->>Sistema: </pre>
Po-sąlyga	Vartotojas gauna norimo formato prototipą parsisiuntimui.

2.3. Nefunkciniai reikalavimai

Pagrindiniai nefunkciniai sistemos reikalavimai nurodyti nuo **2.6 lentelė** iki **2.10 lentelė**. Kiti nefunkciniai sistemos reikalavimai pateikti priede „3 priedas. Nefunkciniai reikalavimai“.

2.6 lentelė. Reikalavimas sistemos išvaizdai (išvaizdos reikalavimas)

Reikalavimas #:	NF-02	Reikalavimo tipas:	V10	PA/FR #:	N/A
Aprašymas:	Sistemos dizaine turėtų dominuoti minimalistinis ir šviesus stilius.				
Pagrindimas:	Pagrindinis vartotojų dėmesys bus skiriamas pradinių duomenų įkėlimui bei apdorotų rezultatų peržiūrai, kuriuose gali būti sąlyginai daug detalių, todėl sistemos dizaino spalvų visuma turėtų būti kuo labiau minimalistinė, o grafinių elementų kiekis neblaškantis dėmesio.				
Šaltinis:	Įmonės MB „Projektavimas“ technologijų vadovas Audrius Marcevičius				
Atitikimo kriterijus:	Reikalavimas bus įgyvendintas jei bent 70 proc. testuotų sistemos naudotojų patvirtins, kad dizainas buvo minimalistinis ir kad jame dominavo šviesios spalvos, kurios neblaškė dėmesio.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	3		
Prioritetas:	Aukštas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-05				

2.7 lentelė. Reikalavimas panaudojamumui (naudojimosi paprastumas)

Reikalavimas #:	NF-04	Reikalavimo tipas:	V11	PA/FR #:	N/A
Aprašymas:	Svarbiausi veiksmai turi būti matomi visur.				
Pagrindimas:	Vartotojai galės greičiau įgyvendinti užduotis sistemoje, jei jas galės pasiekti daug paprasčiau ir greičiau. Jeigu bus pagrindinės navigacinės nuorodos, kurios bus aiškiai matomos visur, tuomet vartotojas galės greitai prirasti kaip greitai naviguoti tarp skirtingų sistemos funkcionalumų.				
Šaltinis:	Ieva Jakšaitytė				
Atitikimo kriterijus:	Reikalavimas bus įgyvendintas jeigu vartotojas bet kurioje sistemos vietoje matys pagrindines nuorodas. Užsakovas matys nuorodą į jo projektus. Sistemos projektuotojas, projektų vadovas ir administratorius matys projektus, prototipus, prototipų generavimą. O administratorius dar matys nustatymus.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	5		
Prioritetas:	Aukštas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-08				

2.8 lentelė. Reikalavimas panaudojamumui (mokymosi reikalavimas)

Reikalavimas #:	NF-05	Reikalavimo tipas:	V11	PA/FR #:	N/A
Aprašymas:	Sistemų projektuotojai ir projektų vadovai turėtų nesunkiai išmokyti naudotis sistema.				
Pagrindimas:	Sistema galės naudotis bet kurie įmonės sistemų projektuotojai ir projektų vadovai, jie nebūtinai turės galimybę sudalyvauti mokymuose, todėl sistema turi būti sukurta taip, kad jiems būtų nesunku suprasti kaip atlikti įvairias užduotis joje.				
Šaltinis:	Ieva Jakšaitytė				
Atitikimo kriterijus:	Reikalavimas bus įgyvendintas jeigu iš testuotų vartotojų bent 80 proc. jų sugebės suprasti ir įgyvendinti bet kurią galimą užduotį sistemoje per greičiau nei 5 min. (po 5 min. kiekvienai atskirai užduočiai išsiaiškinti).				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	4		
Prioritetas:	Aukštas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-09				

2.9 lentelė. Reikalavimas efektyvumui ir našumui (išplečiamumo reikalavimas)

Reikalavimas #:	NF-10	Reikalavimo tipas:	V12	PA/FR #:	N/A
Aprašymas:	Pasirinkti komponentais paremtą architektūrą, kuri leistų lengvai plėsti sistemą pasibaigus projektui.				
Pagrindimas:	Ateityje sistema gali būti pritaikoma veikti papildomose platformose, taip pat, sistemos naudotojai gali norėti papildomo funkcionalumo, kurį būtų lengva įgyvendinti, jeigu sistema būtų paremta komponentų pagrindu.				
Šaltinis:	Įmonės MB „Projektavimas“ technologijų vadovas Audrius Marcevičius				

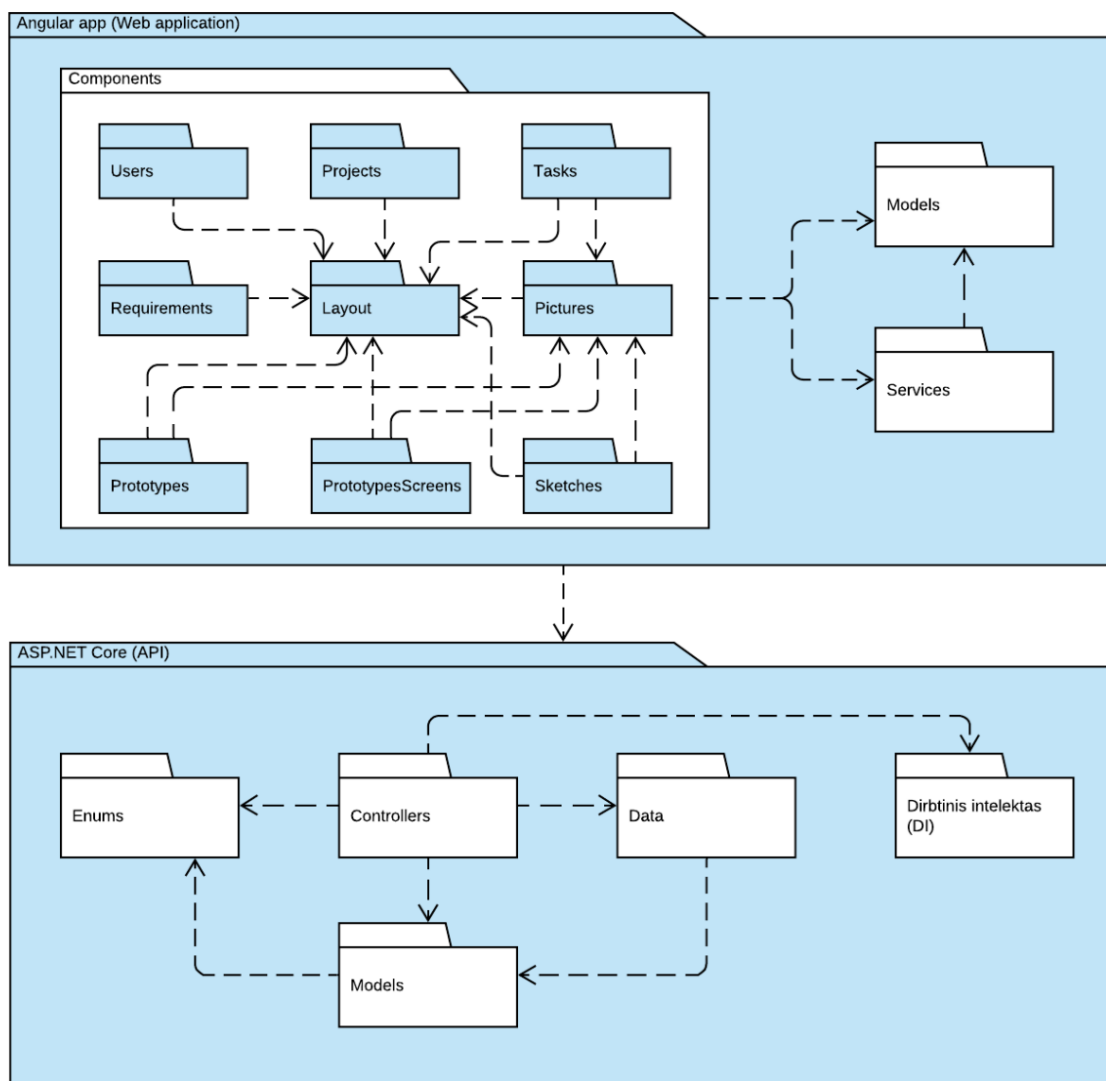
Atitikimo kriterijus:	Pasirinkta sistemos architektūra paremta komponentais.		
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	4
Prioritetas:	Aukštas	Konfliktai:	Nėra
Istorija:	Sukurta 2020-03-09		

2.10 lentelė. Pritaikymo kitoms platformoms (angl. *portability*) reikalavimas

Reikalavimas #:	NF-13	Reikalavimo tipas:	V14	PA/FR #:	N/A
Aprašymas:	Pasirinkti tokią architektūrą, kuri ateityje leistų sukurti mobiliąją programėlę.				
Pagrindimas:	Jeigu sistemos veikimo greitis ir tikslumas pasiteisins, bus norima prototipų generavimą iš eskizų padaryti dar lankstesnį leidžiant fotografuoti ir kelti tiesiai iš mobilios programėlės telefone.				
Šaltinis:	Ieva Jakšaitytė				
Atitikimo kriterijus:	Atitiks kriterijus jei veiks ant kompiuterių bei telefonų su Android arba iOS operacine sistema.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	2		
Prioritetas:	Žemas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-09				

2.4. Sistemą sudarantys paketai

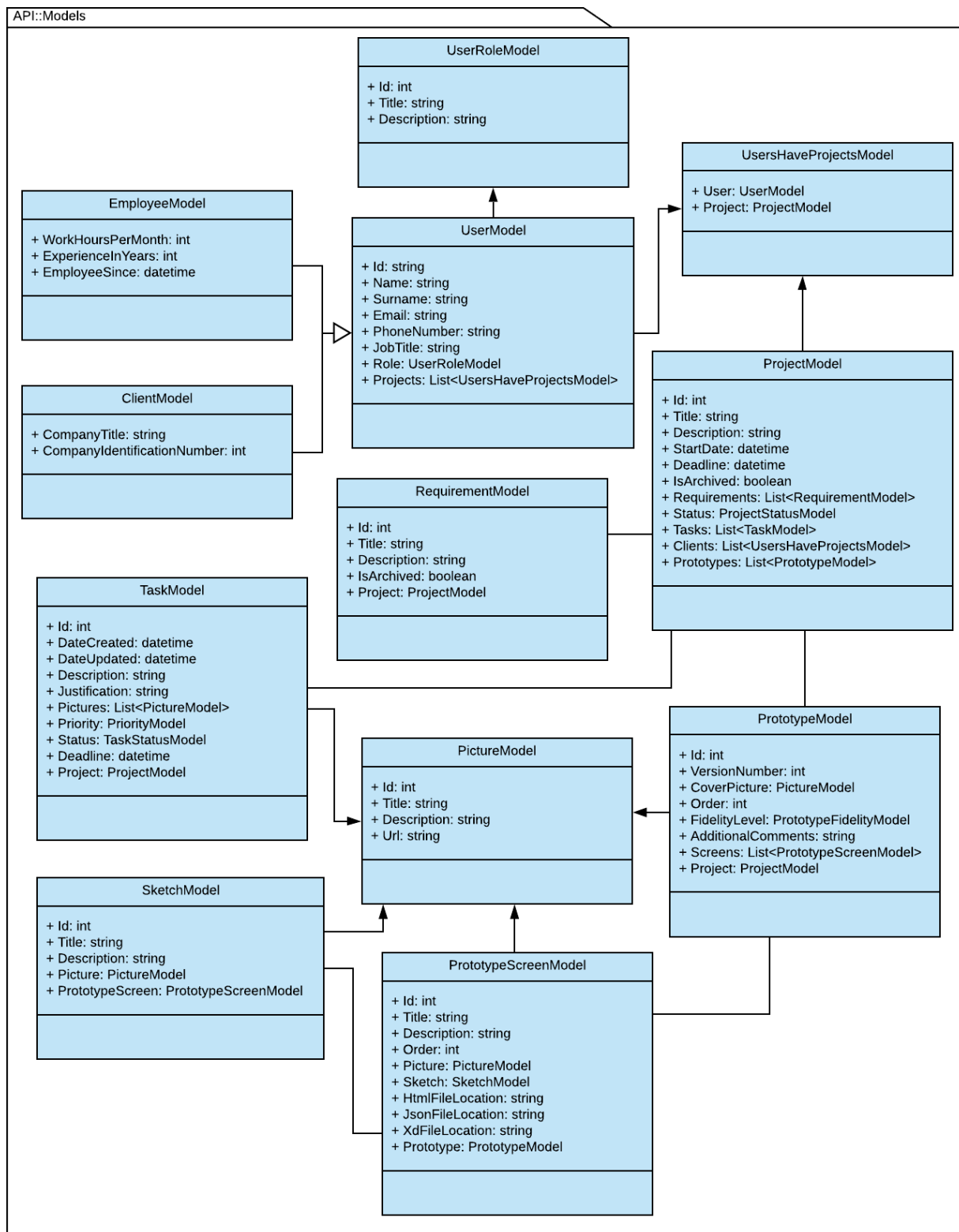
Kuriamo projekto architektūros suskaidymą į paketus galima matyti 2.3 pav. paketų diagramoje. Projektas išskaidytas į dvi pagrindines dalis: API; saityno programa. API įgyvendinta su *ASP.NET Core* karkasu ir atlieka visus pagrindinius veiksmus, bei yra pasiekama iš išorės, kad į ją galėtų kreiptis vartotojams patogios aplikacijos, jame yra dirbtinio intelekto dalis. Vartotojai su sistema gali dirbti per saityno programą, kuri įgyvendinta su *Angular* karkasu, bei kurioje vartotojai gali atlikti visus norimus veiksmus, o pati saityno programa komunikuoja su API.



2.3 pav. Sistemą sudarantys paketai

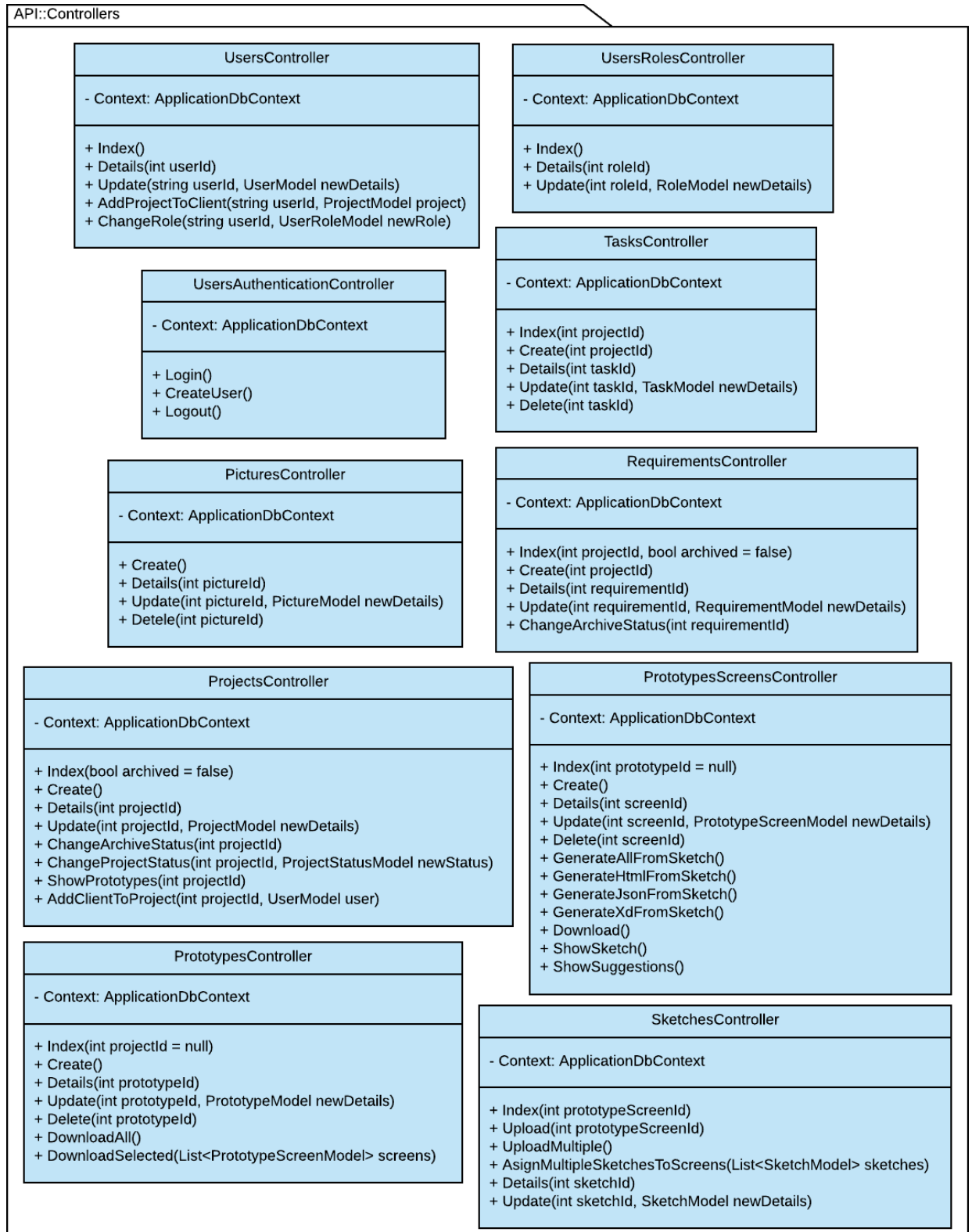
API paketo detalizavimas

Paketo *API::Models* klasių diagrama matoma 2.4 pav.. Šiame pakete yra pateikiami duomenų modeliai, kurie naudojami visoje sistemoje. Šios klasės neatlieka jokių veiksmų, nes yra skirtos tik duomenų modelių aprašymui, kaip tai yra nurodyta pagal MVC principus.



2.4 pav. Paketo *API::Models* klasių diagrama

Paketo *API::Controllers* klasių diagrama matoma **2.5 pav.**. Šiame pakete pateikiamos klasės, kurios atlieka visus pagrindinius sistemos veiksmus, tokius kaip projekto sukūrimas, prototipo sugeneravimas iš nuskaityto eskizo ir t. t. Visose šio paketo klasėse įgyvendinta priklausomybių „injekcija“ (angl. *Dependency Injection*), kas leidžia patogiai panaudoti bet kurios klasės, esančios *API::Models* pakete informaciją, o modeliai su šiomis klasėmis surišti per atributą *Context*.

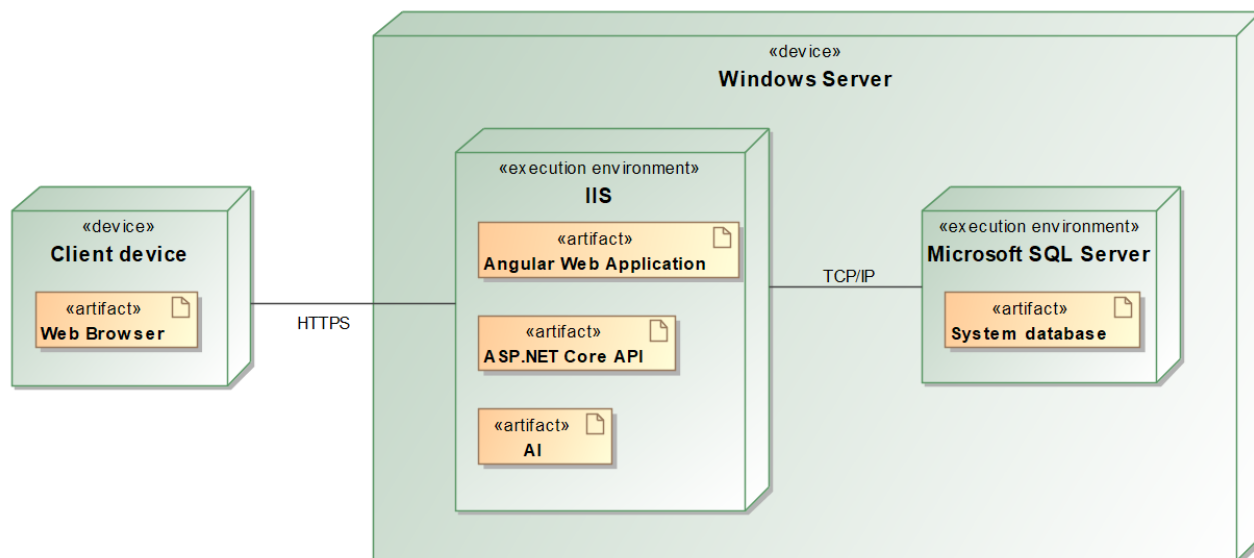


2.5 pav. Paketo *API::Controllers* klasių diagrama

Likusių paketų diagramų detalizavimas pateiktas priede „4 priedas. Detalizuotos paketų diagramos“.

2.5. Sistemos diegimo vaizdas

Sistemos diegimo vaizdas matomas 2.6 pav..

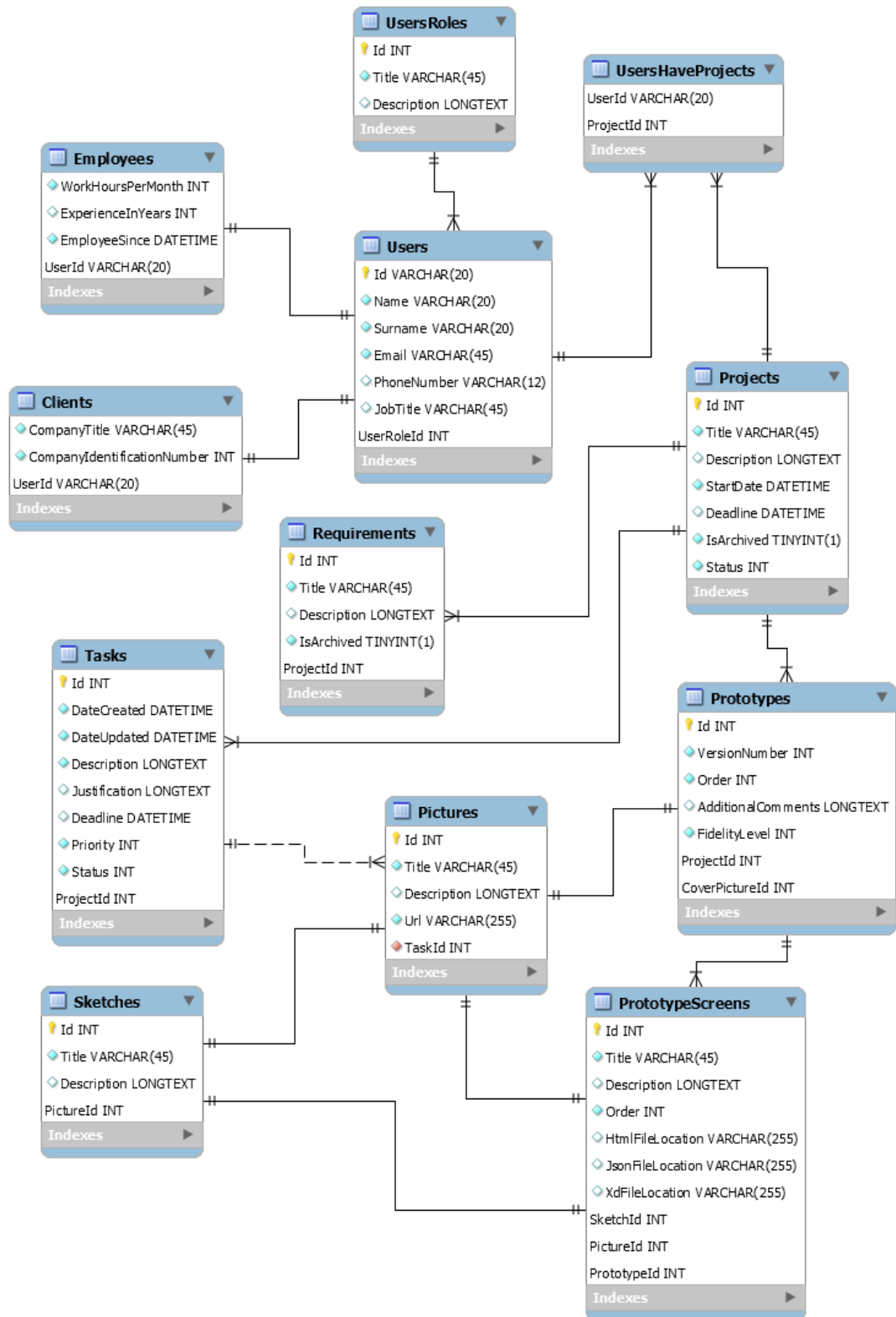


2.6 pav. Sistemos diegimo vaizdas

Sukurta programinė įranga veikia nutolusiame serveryje, kurią vartotojas gali pasiekti internetu naršyklės pagalba. Yra naudojamas serveris su *Windows Server* operacine sistema. Šiame serveryje yra naudojamos dvi paleidimo aplinkos, tai *IIS* ir *Microsoft SQL Server*. *IIS* bus naudojamas paleisti tris skirtingas programinės įrangos dalis, tai *Angular Web Application*, *ASP.NET Core API* ir *AI*, kurios dirba tarpusavyje, taip pat, prireikus jos gali kreiptis į *Microsoft SQL Server* duomenų bazę.

2.6. Duomenų vaizdas

Duomenų bazės modelio schema matoma 2.7 pav.. Šioje schemoje pagrindinės lentelės yra „Projects“, „Prototypes“, „PrototypesScreens“ ir „Sketches“, kurios yra skirtos saugoti informaciją apie projekto, jų projektavime numatytus langus bei pačius įkeltus eskizus.



2.7 pav. Duomenų bazės modelio schema

3. Tyrimo dalis

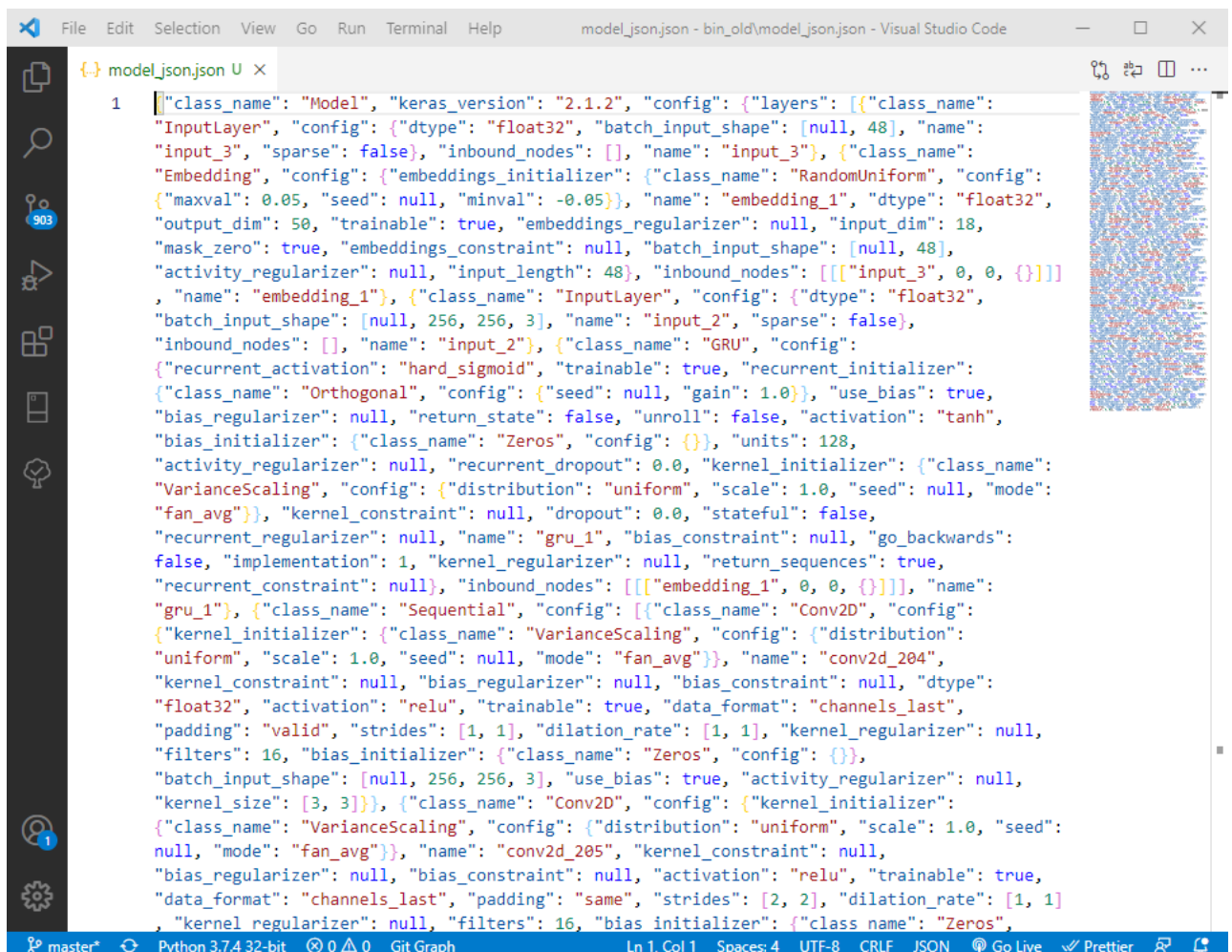
3.1. Išplėsto modelio tyrimas

Buvo tyrinėjami *pix2code*, *SketchCode* ir kiti susijusių projektų dirbtinio intelekto modeliai. Šie modeliai veikia labai panašiai, todėl šiame poskyryje pateikiama bendrinė informacija, tinkanti visiems tokiems sprendimams.

3.1.1. Projekto dirbtinio intelekto dalies apžvalga

Pagrindiniai failai

Dirbtinio intelekto modelis saugomas *model_json.json* faile (3.1 pav.). Permokant modelį yra sukuriamas naujas **.json* formato failas, kurį vėliau galima naudoti norint atpažinti informaciją, esančią eskize ir taip sukurti galutinį **.html* ar kito formato failą. Apmokyto modelio svoriai saugomi *weights.h5* faile.



```
1 [{"class_name": "Model", "keras_version": "2.1.2", "config": {"layers": [{"class_name": "InputLayer", "config": {"dtype": "float32", "batch_input_shape": [null, 48], "name": "input_3", "sparse": false, "inbound_nodes": [], "name": "input_3"}, {"class_name": "Embedding", "config": {"embeddings_initializer": {"class_name": "RandomUniform", "config": {"maxval": 0.05, "seed": null, "minval": -0.05}}, "name": "embedding_1", "dtype": "float32", "output_dim": 50, "trainable": true, "embeddings_regularizer": null, "input_dim": 18, "mask_zero": true, "embeddings_constraint": null, "batch_input_shape": [null, 48], "activity_regularizer": null, "input_length": 48, "inbound_nodes": [[["input_3", 0, 0, {}]]], "name": "embedding_1"}, {"class_name": "InputLayer", "config": {"dtype": "float32", "batch_input_shape": [null, 256, 256, 3], "name": "input_2", "sparse": false, "inbound_nodes": [], "name": "input_2"}, {"class_name": "GRU", "config": {"recurrent_activation": "hard_sigmoid", "trainable": true, "recurrent_initializer": {"class_name": "Orthogonal", "config": {"seed": null, "gain": 1.0}}, "use_bias": true, "bias_regularizer": null, "return_state": false, "unroll": false, "activation": "tanh", "bias_initializer": {"class_name": "Zeros", "config": {}}, "units": 128, "activity_regularizer": null, "recurrent_dropout": 0.0, "kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "scale": 1.0, "seed": null, "mode": "fan_avg"}}, "kernel_constraint": null, "dropout": 0.0, "stateful": false, "recurrent_regularizer": null, "name": "gru_1", "bias_constraint": null, "go_backwards": false, "implementation": 1, "kernel_regularizer": null, "return_sequences": true, "recurrent_constraint": null, "inbound_nodes": [[["embedding_1", 0, 0, {}]]], "name": "gru_1"}, {"class_name": "Sequential", "config": [{"class_name": "Conv2D", "config": {"kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "scale": 1.0, "seed": null, "mode": "fan_avg"}}, "name": "conv2d_204", "kernel_constraint": null, "bias_regularizer": null, "bias_constraint": null, "dtype": "float32", "activation": "relu", "trainable": true, "data_format": "channels_last", "padding": "valid", "strides": [1, 1], "dilation_rate": [1, 1], "kernel_regularizer": null, "filters": 16, "bias_initializer": {"class_name": "Zeros", "config": {}}, "batch_input_shape": [null, 256, 256, 3], "use_bias": true, "activity_regularizer": null, "kernel_size": [3, 3]}, {"class_name": "Conv2D", "config": {"kernel_initializer": {"class_name": "VarianceScaling", "config": {"distribution": "uniform", "scale": 1.0, "seed": null, "mode": "fan_avg"}}, "name": "conv2d_205", "kernel_constraint": null, "bias_regularizer": null, "bias_constraint": null, "activation": "relu", "trainable": true, "data_format": "channels_last", "padding": "same", "strides": [2, 2], "dilation_rate": [1, 1], "kernel_regularizer": null, "filters": 16, "bias_initializer": {"class_name": "Zeros",
```

3.1 pav. Dirbtinio intelekto modelio failas

Modelio apmokymui yra reikalinga, kad kiekvienas eskizas būtų pateiktas naudojant **.png* formato nuotrauką (3.2 pav.), bei eskize esančių elementų aprašą **.gui* formato faile (3.3 pav.). Originaliame duomenų rinkinyje pateikta 1700 vienetų eskizų, kur kiekvienas eskizas sudarytas iš atvaizdo ir aprašo failų.



© Tony Beltramelli 2017

3.2 pav. Originalaus duomenų rinkinio eskizo nuotrauka

Eskizo nuotraukos, kuri matoma 3.2 pav., aprašas pavaizduotas 3.3 pav.. Šis eskizas yra sudarytas iš viršutinės juostos, trijų eilių, kiekvienoje eilėje atitinkamo kiekio stulpelių bei turinio, pateikto kaip antraštė, paprastas tekstas ar mygtukas.

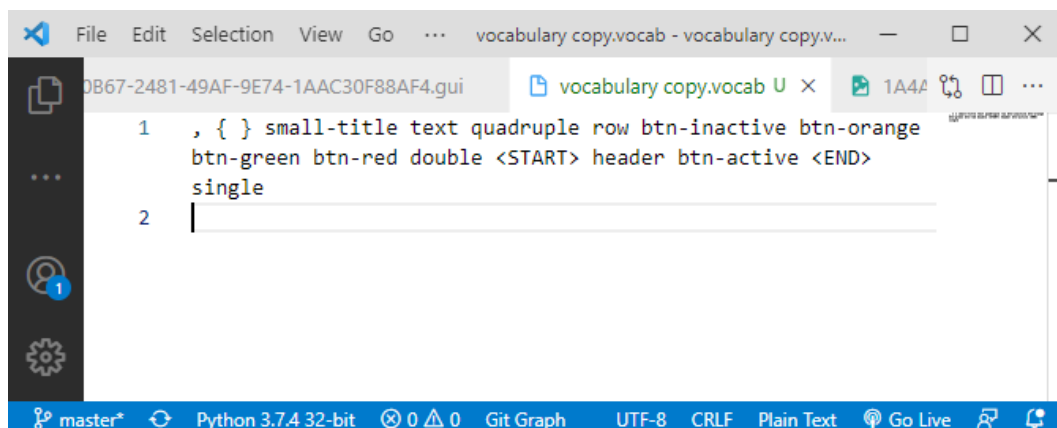
```

File Edit Selection ... 1A4A0B67-2481-49AF-9E74-1AAC30F88AF4.gui - data\all_data\1A4A...
example1.gui U 1A4A0B67-2481-49AF-9E74-1AAC30F88AF4.gui x 1A4A0B67-2481-49...
1 header {
2 btn-inactive, btn-inactive, btn-inactive
3 }
4 row {
5 double {
6 small-title, text, btn-orange
7 }
8 double {
9 small-title, text, btn-orange
10 }
11 }
12 row {
13 single {
14 small-title, text, btn-orange
15 }
16 }
17 row {
18 quadruple {
19 small-title, text, btn-orange
20 }
21 quadruple {
22 small-title, text, btn-orange
23 }
24 quadruple {
25 small-title, text, btn-orange
26 }
27 quadruple {
28 small-title, text, btn-orange
29 }
30 }
31 }

```

3.3 pav. Originalaus duomenų rinkinio eskizo aprašas

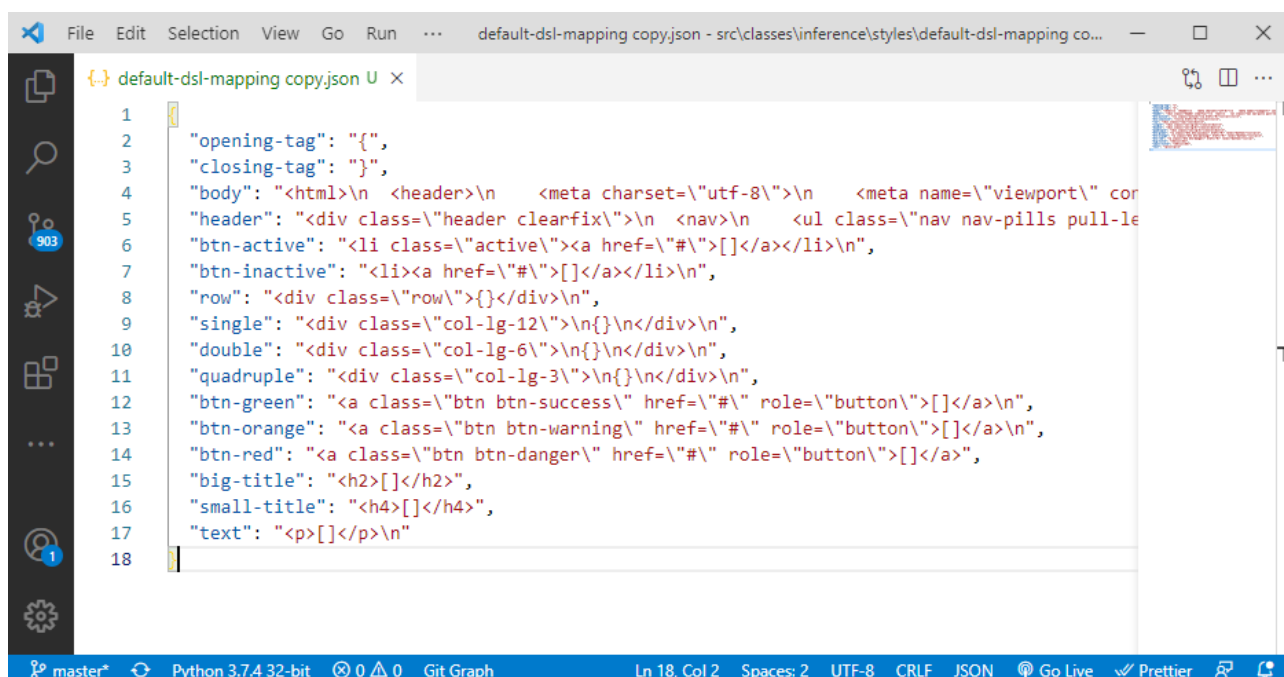
Eskizo aprašo *.gui failė gali bŭti naudojami tik iŝ anksto numatyti ųymėjimai, kurie pateikti ųodyno *vocabulary.vocab* failė (3.4 pav.). Originaliame rinkinyje numatyti 12 iŝdėstymo ir turinio ųymėjimŭ bei 5 pagalbiniai ųymėjimai.



```
1 , { } small-title text quadruple row btn-inactive btn-orange
  btn-green btn-red double <START> header btn-active <END>
  single
2 |
```

3.4 pav. Originalaus duomenŭ rinkinio ųodynas

Po atlikto eskizo nuskaitymo ir jo aprašo sugeneravimo *.gui formatu, ŝis eskizas gali bŭti transformuojamas iŝ *.html ar kito formato failŭ, jam pritaikant norimŭ ŝablonŭ (angl. *theme*). Originalaus duomenŭ rinkinio transformavimo iŝ saityno programŭ ųodynas matomas 3.5 pav..

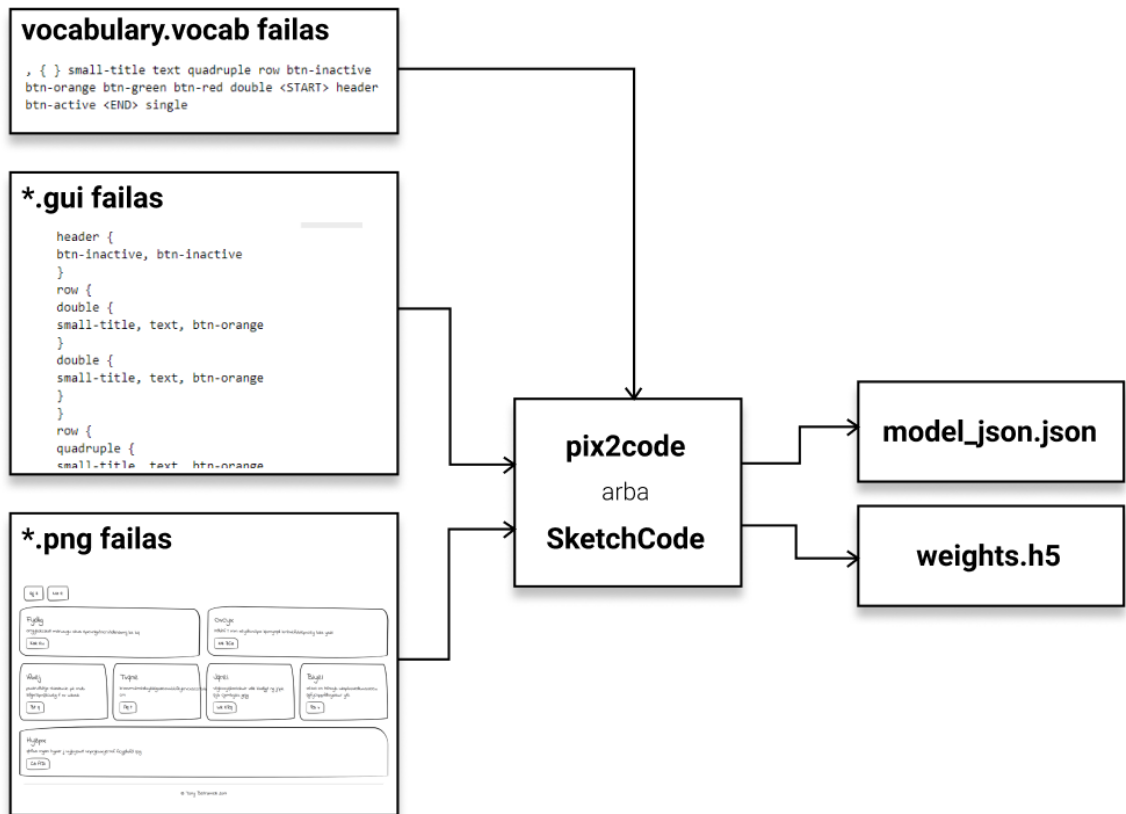


```
1
2 "opening-tag": "{",
3 "closing-tag": "}",
4 "body": "<html>\n <header>\n <meta charset=\"utf-8\">\n <meta name=\"viewport\" con
5 "header": "<div class=\"header clearfix\">\n <nav>\n <ul class=\"nav nav-pills pull-le
6 "btn-active": "<li class=\"active\"><a href=\"#\">[]</a></li>\n",
7 "btn-inactive": "<li><a href=\"#\">[]</a></li>\n",
8 "row": "<div class=\"row\">{}</div>\n",
9 "single": "<div class=\"col-lg-12\">\n{}</div>\n",
10 "double": "<div class=\"col-lg-6\">\n{}</div>\n",
11 "quadruple": "<div class=\"col-lg-3\">\n{}</div>\n",
12 "btn-green": "<a class=\"btn btn-success\" href=\"#\" role=\"button\">[]</a>\n",
13 "btn-orange": "<a class=\"btn btn-warning\" href=\"#\" role=\"button\">[]</a>\n",
14 "btn-red": "<a class=\"btn btn-danger\" href=\"#\" role=\"button\">[]</a>\n",
15 "big-title": "<h2>[]</h2>\n",
16 "small-title": "<h4>[]</h4>\n",
17 "text": "<p>[]</p>\n"
18
```

3.5 pav. Originalaus duomenŭ rinkinio transformavimo iŝ saityno programŭ ųodynas

Modelio apmokymas

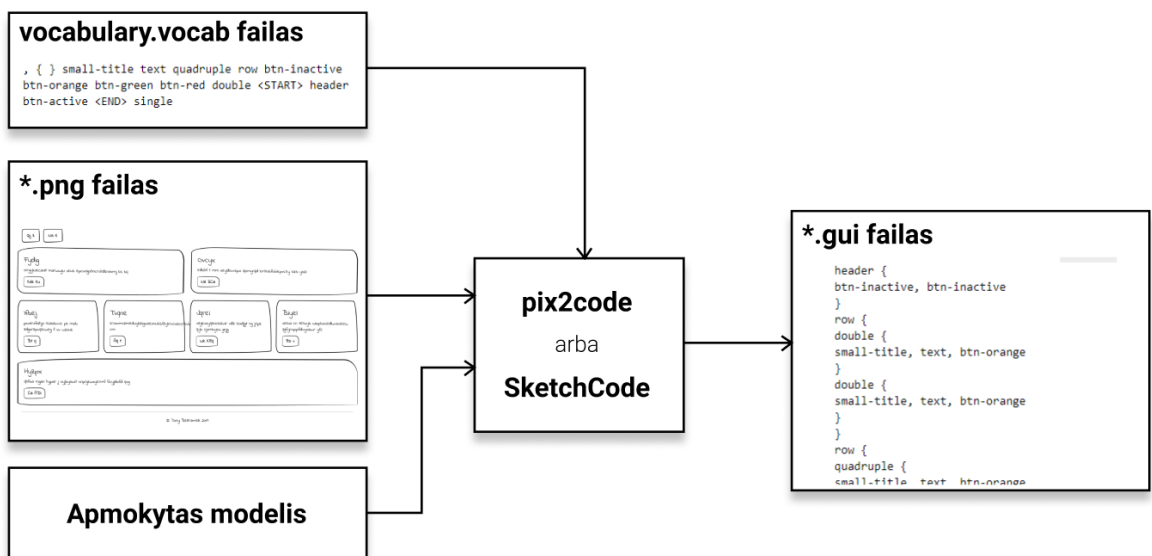
Modelio apmokymui reikia eskizŭ informacijos, pateiktos *.gui ir *.png failuose. Apmokymas gali bŭti vykdomas per *pix2code* arba *SketchCode* sprendimŭ. Galutinis apmokytas modelis saugomas *model_json.json* failė, o jo svoriai *weights.h5* failė. Visas modelio apmokymo procesas pavaizduotas 3.6 pav..



3.6 pav. Modelio apmokymo diagrama

Eskize esančios informacijos atpažinimas

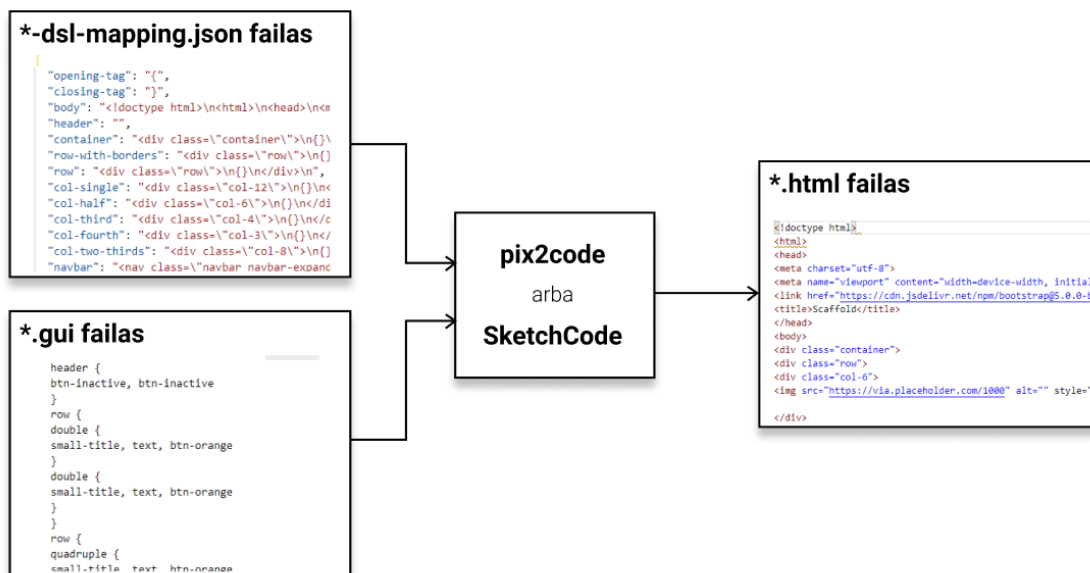
Norint atpažinti informaciją esančią eskize, iš anksto turi būti apmokytas modelis, kuris galėtų interpretuoti eskizo atvaizde esančius simbolius ar kitus žymėjimus ir parengtas žymėjimų žodyno *vocabulary.vocab* failas. Norimo apdoroti eskizo atvaizdas pateikiamas **.png* formatu, gaunami rezultatai yra **.gui* formato. Eskize esančios informacijos atpažinimo diagrama matoma 3.7 pav..



3.7 pav. Eskize esančios informacijos atpažinimo diagrama

Eskizo aprašo transformavimas į kitą pasirinktą formatą

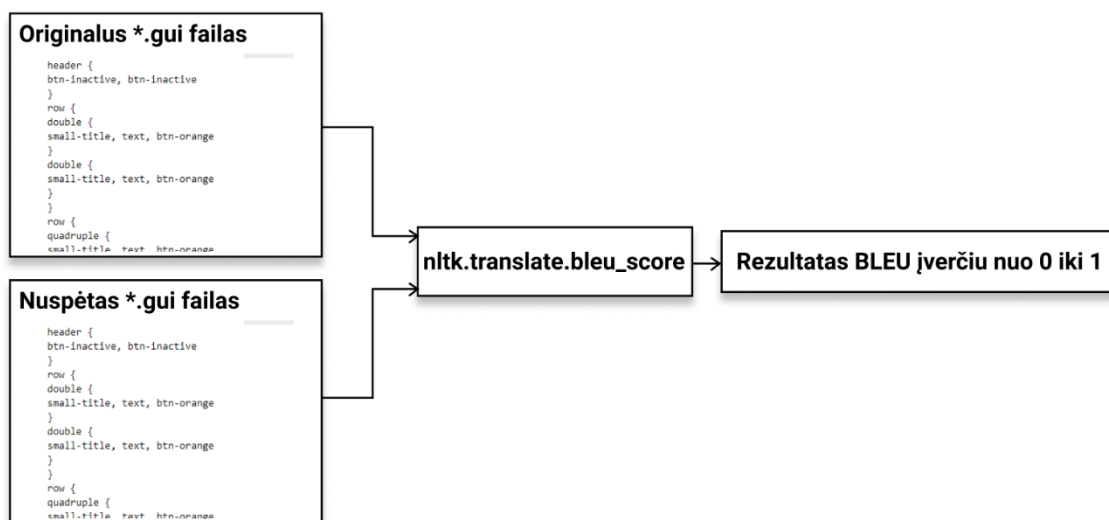
Atpažinus informaciją esančią eskize, eskizo aprašą galima transformuoti į pasirinktos aplinkos kodą, šiuo atveju tai yra **.html* formato failas. Transformavimui yra reikalingas eskizo aprašo **.gui* failas ir aprašo transformavimo į pasirinktą aplinką žodyno failas. Eskizo aprašo transformavimo į **.html* diagrama matoma **3.8 pav.**



3.8 pav. Eskizo aprašo transformavimo į **.html* diagrama

Atpažinto eskizo tikslumo nustatymas

Norint įvertinti eskizo atpažinimo tikslumą, tam buvo panaudotas BLEU įvertis (angl. *BLEU score*). Šiam įverčiui panaudoti reikalingas originalaus eskizo **.gui* failo informacija, bei atpažinto eskizo **.gui* informacija (**3.9 pav.**). Jeigu šiuose failuose informacija atitinka pagal kiekvieną elementą, tai BLEU įvertis bus lygus 1, o jei niekas neatitiks bus lygus 0. Su šiuo įverčiu galima įvertinti kaip tiksliai modelis sugeba atpažinti eskizus ir apibūdinti juose esančią informaciją.



3.9 pav. Atpažinto eskizo tikslumo nustatymas

3.1.2. Duomenų paruošimas modelio apmokymui

Norint paruošti duomenis modelio apmokymui reikia turėti omenyje tai, kad kiekvienas eskizas turės turėti atvaizdo nuotrauką *.png formatu bei eskize esančių elementų aprašą *.gui formatu. Aprašo failą sugeneruoti ir įrašyti galima tiesiai, nes jame esantis turinys yra paprasčiausia tekstinė informacija, tačiau dėl nuotraukos sugeneravimo nuspręsta kurti tarpinį *.html formato failą ir tuomet jį paversti į nuotrauką. Taip galima orientuotis į saityno technologijų veikimą, o ne taškų dėliojimą nuotraukoje, galima padaryti tinklalapio išvaizdą panašią į eskizą naudojant CSS stilių rašymo kalbą. Taip pat, yra parengta įvairių priemonių, kurios turimą HTML kodą leidžia atvaizduoti panašų į piešinį, viena tokių – *PaperCSS*, kuri ir buvo panaudota kuriant HTML failą su atitinkamu stiliumi.

Abu failų formatai (*.gui ir *.html) generuojami vienu metu (3.10 pav.) ir tuomet generuojamas paveikslukas iš gauto *.html failo. Šių failų generavimas vyksta *JavaScript* programavimo kalbos pagalba, kodą vykdant per *NodeJS* serverį, nes taip sąlyginai paprastai galima sukurti saityno ar kito turinio ir jį pritaikyti kaip reikia, o taip pat galima kurti naujus failus pasirinktame kataloge.

Programa atsitiktinai renkasi kiek turinio sukurti ir kokias turinio variacijas panaudoti. Vienu metu yra įsimenamas *.gui kodas ir *.html kodas. Programos pavyzdys kaip yra generuojamas *.gui ir *.html stulpelių ir juose esančio turinio kodas, matomas 3.10 pav..

```
function generateColumns(colsVar, colType, colContent) {
  let guiContent = colContent.join(', ');
  let htmlContent = convertColContentToHtml(colContent);

  switch (colsVar) {
    case 1:
      if (colType == 'gui') {
        return `col-single {
          ${ guiContent }
        }`;
      }
      else if (colType == 'web') {
        return `

3.10 pav. Stulpelio ir jo turinio generavimas *.gui ir *.html formatais



42


```

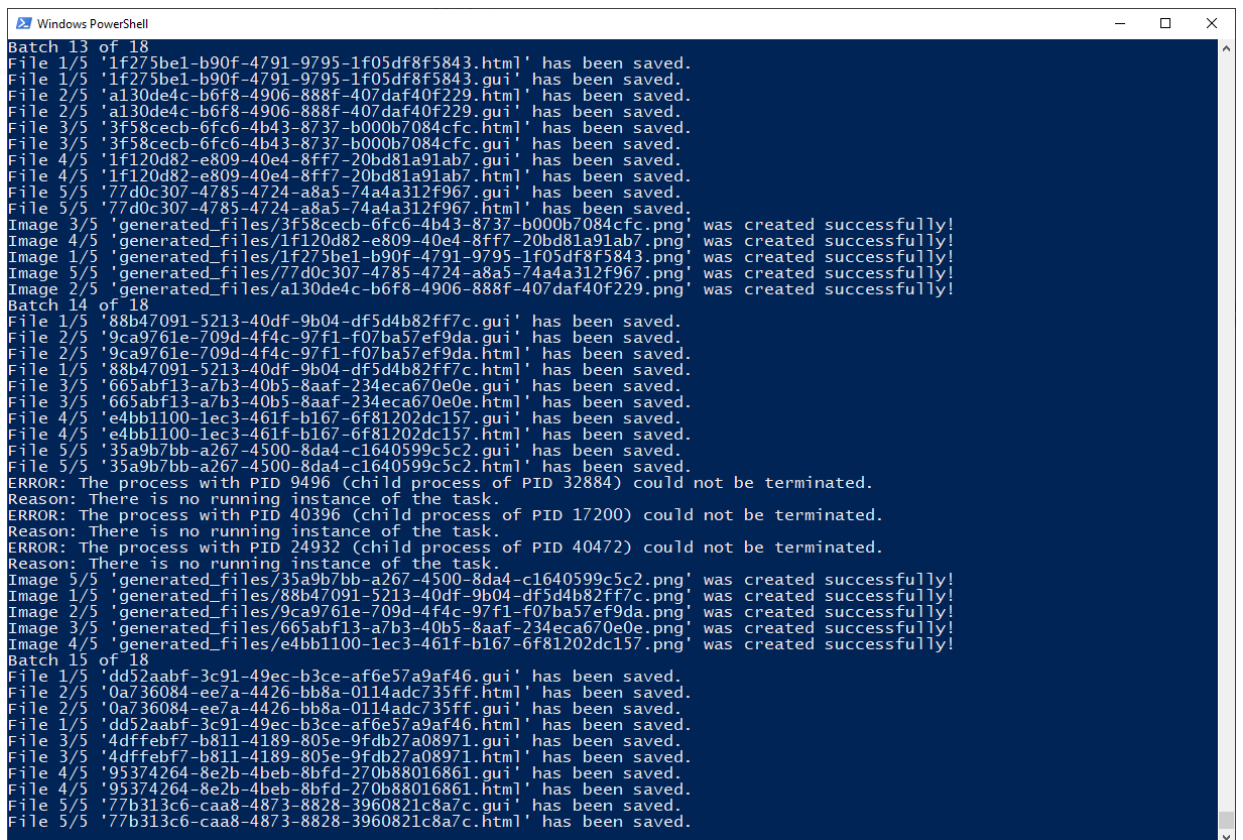
Paruošto kodo rašymas į failus matomas **3.11 pav.**, ten pat matoma, kad HTML failo konvertavimas į paveiksluką vyksta po **.html* failo sukūrimo.

```
fs.writeFile(`generated_files/${fileName}.html`, htmlTemplate, (err) => {
  if (err) {
    throw err;
  }
  console.log(`File ${i+1}/${filesCount} '${fileName}.html' has been saved.`);
  nodeHtmlToImage({
    output: `generated_files/${fileName}.png`,
    html: htmlTemplate
  })
  .then(() => console.log(`Image ${i+1}/${filesCount} 'generated_files/${fileName}.png'
    was created successfully!`));
});

fs.writeFile(`generated_files/${fileName}.gui`, guiCode, (err) => {
  if (err) {
    throw err;
  }
  console.log(`File ${i+1}/${filesCount} '${fileName}.gui' has been saved.`);
});
```

3.11 pav. Sugeneruoto eskizo informacijos įrašymas į failus ir paveiksluko sukūrimas

Paleidus duomenų generavimo programą (**3.12 pav.**) galima rinktis kiek eskizų turėtų būti sukurta. Programa generuos po 5 paveikslukus vienos iteracijos metu, dėl failų rašymo į sistemą apribojimų. Deja, daugiausiai laiko užtrunka konvertavimas iš **.html* failo į **.png* failą, todėl sukurti 2 tūkst. eskizų užtrunka apie valandą.

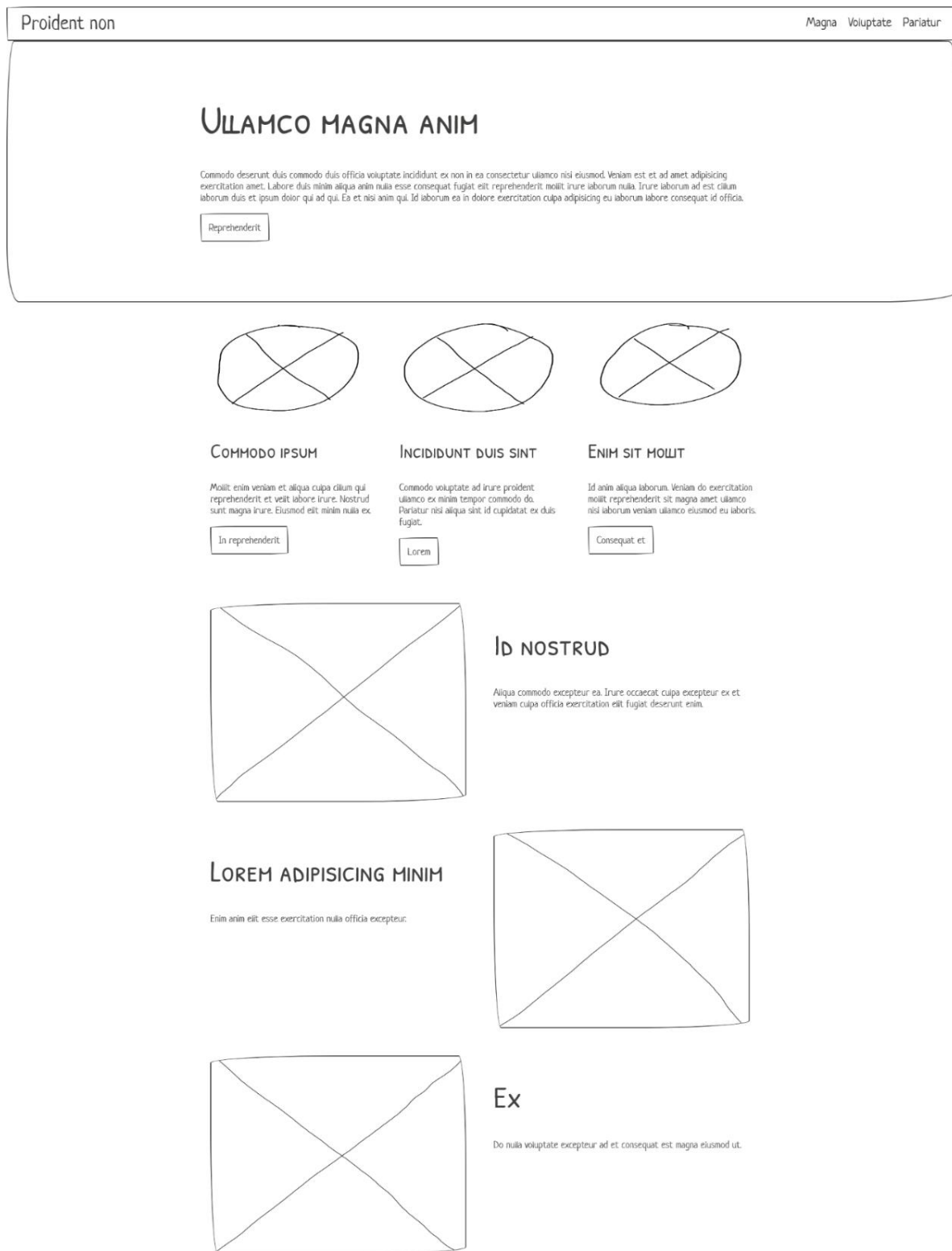


```
Windows PowerShell
Batch 13 of 18
File 1/5 '1f275be1-b90f-4791-9795-1f05df8f5843.html' has been saved.
File 1/5 '1f275be1-b90f-4791-9795-1f05df8f5843.gui' has been saved.
File 2/5 'a130de4c-b6f8-4906-888f-407daf40f229.html' has been saved.
File 2/5 'a130de4c-b6f8-4906-888f-407daf40f229.gui' has been saved.
File 3/5 '3f58cecb-6fc6-4b43-8737-b000b7084cfc.html' has been saved.
File 3/5 '3f58cecb-6fc6-4b43-8737-b000b7084cfc.gui' has been saved.
File 4/5 '1f120d82-e809-40e4-8ff7-20bd81a91ab7.html' has been saved.
File 4/5 '1f120d82-e809-40e4-8ff7-20bd81a91ab7.gui' has been saved.
File 5/5 '77d0c307-4785-4724-a8a5-74a4a312f967.html' has been saved.
File 5/5 '77d0c307-4785-4724-a8a5-74a4a312f967.gui' has been saved.
Image 3/5 'generated_files/3f58cecb-6fc6-4b43-8737-b000b7084cfc.png' was created successfully!
Image 4/5 'generated_files/1f120d82-e809-40e4-8ff7-20bd81a91ab7.png' was created successfully!
Image 1/5 'generated_files/1f275be1-b90f-4791-9795-1f05df8f5843.png' was created successfully!
Image 5/5 'generated_files/77d0c307-4785-4724-a8a5-74a4a312f967.png' was created successfully!
Image 2/5 'generated_files/a130de4c-b6f8-4906-888f-407daf40f229.png' was created successfully!
Batch 14 of 18
File 1/5 '88b47091-5213-40df-9b04-df5d4b82ff7c.gui' has been saved.
File 2/5 '9ca9761e-709d-4f4c-97f1-f07ba57ef9da.gui' has been saved.
File 2/5 '9ca9761e-709d-4f4c-97f1-f07ba57ef9da.html' has been saved.
File 1/5 '88b47091-5213-40df-9b04-df5d4b82ff7c.html' has been saved.
File 3/5 '665abf13-a7b3-40b5-8aaf-234eca670e0e.gui' has been saved.
File 3/5 '665abf13-a7b3-40b5-8aaf-234eca670e0e.html' has been saved.
File 4/5 'e4bb1100-1ec3-461f-b167-6f81202dc157.gui' has been saved.
File 4/5 'e4bb1100-1ec3-461f-b167-6f81202dc157.html' has been saved.
File 5/5 '35a9b7bb-a267-4500-8da4-c1640599c5c2.gui' has been saved.
File 5/5 '35a9b7bb-a267-4500-8da4-c1640599c5c2.html' has been saved.
ERROR: The process with PID 9496 (child process of PID 32884) could not be terminated.
Reason: There is no running instance of the task.
ERROR: The process with PID 40396 (child process of PID 17200) could not be terminated.
Reason: There is no running instance of the task.
ERROR: The process with PID 24932 (child process of PID 40472) could not be terminated.
Reason: There is no running instance of the task.
Image 5/5 'generated_files/35a9b7bb-a267-4500-8da4-c1640599c5c2.png' was created successfully!
Image 1/5 'generated_files/88b47091-5213-40df-9b04-df5d4b82ff7c.png' was created successfully!
Image 2/5 'generated_files/9ca9761e-709d-4f4c-97f1-f07ba57ef9da.png' was created successfully!
Image 3/5 'generated_files/665abf13-a7b3-40b5-8aaf-234eca670e0e.png' was created successfully!
Image 4/5 'generated_files/e4bb1100-1ec3-461f-b167-6f81202dc157.png' was created successfully!
Batch 15 of 18
File 1/5 'dd52aabf-3c91-49ec-b3ce-af6e57a9af46.gui' has been saved.
File 2/5 '0a736084-ee7a-4426-bb8a-0114adc735ff.html' has been saved.
File 2/5 '0a736084-ee7a-4426-bb8a-0114adc735ff.gui' has been saved.
File 1/5 'dd52aabf-3c91-49ec-b3ce-af6e57a9af46.html' has been saved.
File 3/5 '4dffebf7-b811-4189-805e-9fdb27a08971.gui' has been saved.
File 3/5 '4dffebf7-b811-4189-805e-9fdb27a08971.html' has been saved.
File 4/5 '95374264-8e2b-4beb-8bfd-270b88016861.gui' has been saved.
File 4/5 '95374264-8e2b-4beb-8bfd-270b88016861.html' has been saved.
File 5/5 '77b313c6-caa8-4873-8828-3960821c8a7c.gui' has been saved.
File 5/5 '77b313c6-caa8-4873-8828-3960821c8a7c.html' has been saved.
```

3.12 pav. Duomenų (eskizų) generavimas

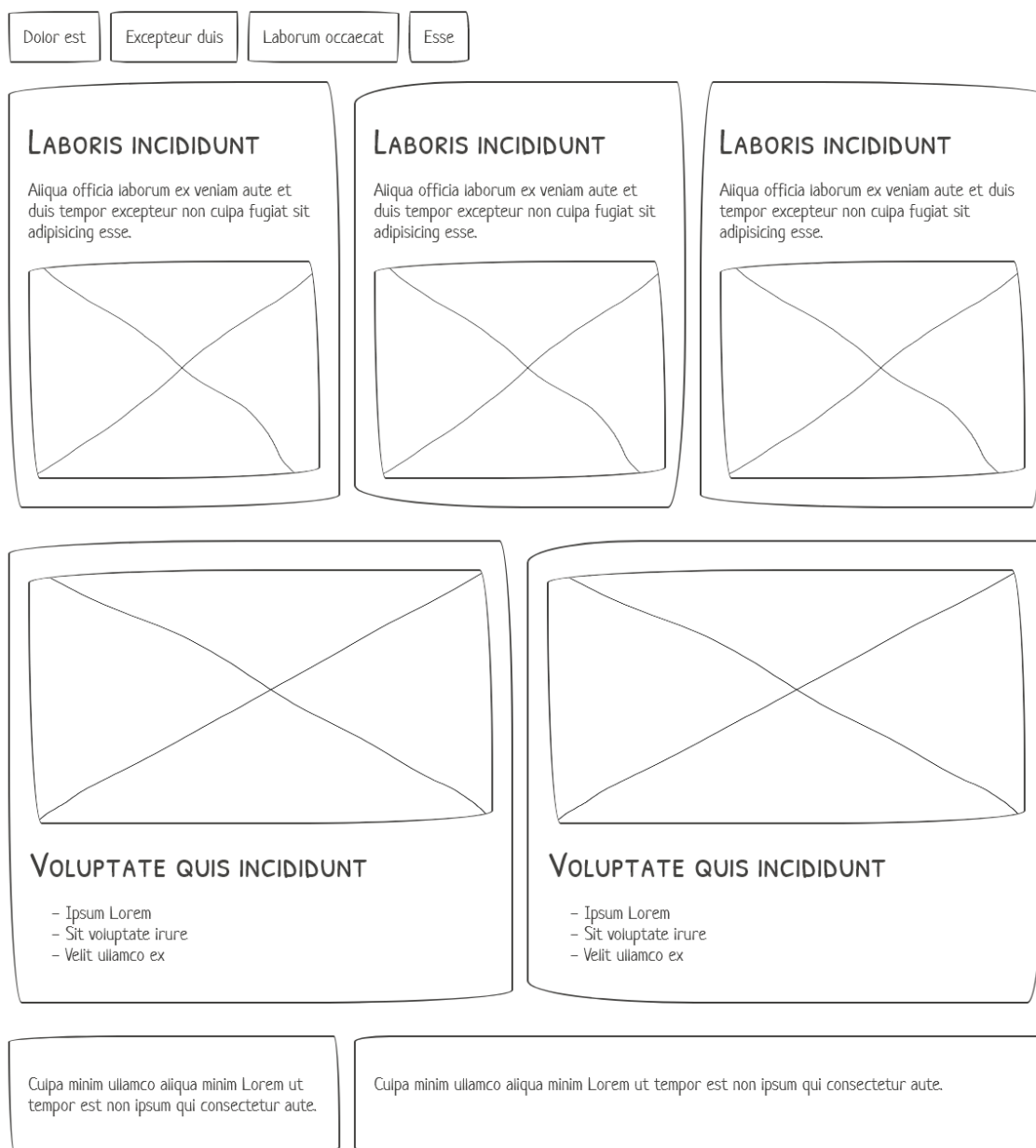
Naujai sukurtuose eskizuose (3.13 pav., 3.14 pav., 3.15 pav.) yra pridėta daugiau turinio išdėstymo variacijų ir daugiau simbolių žymėjimų.

Sudėtingiausiuose eskizuose yra daugiausiai turinio variantų ir turinio pagal kiekį (3.13 pav.). nurodytame eskize galima pastebėti navigacijos juostą, viršutinę tinklalapio dalį (angl. header), nuotraukas (apvalias ir kvadratinės ar stačiakampes), antraštes, paprastą tekstą, mygtukus.



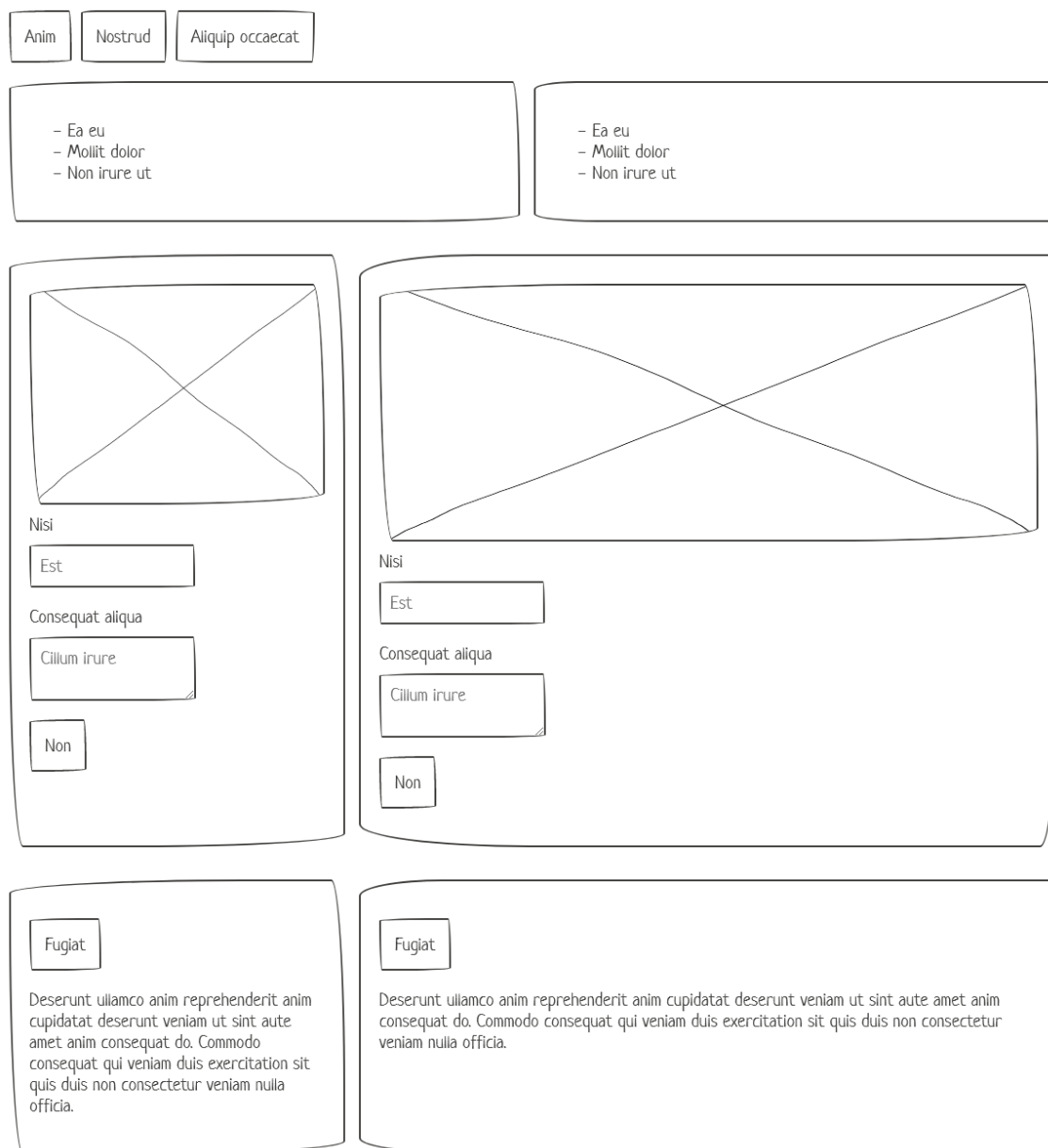
3.13 pav. Naujai sukurto eskizo pavyzdys (1)

Taip pat, eskizai gali būti ir paprastesni, su mažiau turinio (3.14 pav.). Pavaizduotame eskize galima pastebėti, kad yra naudojamos antraštės, paprastas tekstas, mygtukai, nuotraukos ir nerikiuoti sąrašai.



3.14 pav. Naujai sukurto eskizo pavyzdys (2)

Eskizuose galima rasti ir formų (kontaktinių ar kitų), kaip tai matoma **3.15 pav.**



3.15 pav. Naujai sukurto eskizo pavyzdys (3)

Pradinio duomenų rinkinio eskizų palyginimas su naujai sukurtu rinkiniu matomas **3.1 lentelėje**.

3.1 lentelė. Pradinių duomenų rinkinio eskizų ir naujai sukurtų eskizų palyginimas

Palyginimo kriterijus	Pradinių duomenų rinkinio eskizai	Naujai sukurti eskizai
Turinio apribojimas plotyje (turinys dedamas į „konteinerį“)	Ne	Taip
Stulpeliai eilutėje skaidomi po	1, 2, 4	1, 2, 3, 4, 2/3, 1/3
Rėmelių uždėjimas ant eilutėje esančių stulpelių	Ne	Taip, galima be rėmelių arba su rėmeliais
Antraštės	Taip, vieno lygio antraštė	Taip, trijų lygių antraštės
Tekstas	Taip	Taip
Mygtukai	Taip, 5 variacijos	Taip, 1 variacija

Navigacijos juosta	Ne	Taip
Viršutinė tinklalapio dalis (angl. <i>header</i>)	Taip, tačiau neatitinkanti realybės	Taip
Nuotraukos	Ne	Taip, 3 variacijos (stačiakampė, kvadratinė, apvali)
Nerikiuotas sąrašas	Ne	Taip
Forma (pavyzdžiui, kontaktinė forma)	Ne	Taip

3.1.3. Projekto paruošimas modelio apmokymui ir galutinių rezultatų sugeneravimui

Modelio apmokymui be duomenų dar reikia eskizų **.gui* failuose naudojamų žymėjimų žodyno informacijos. Tokia informacija yra saugoma *vocabulary.vocab* faile, todėl šis failas turi būti atitinkamai pakeičiamas, kad jame būtų išvardinti visi **.gui* failuose naudojami žymėjimai bei palikti pagalbiniai žymiai. Naujai sukurto eskizų rinkinio žodyno failas matomas **3.16 pav.**

```

You, 43 minutes ago | 1 author (You)
1 , { } <START> <END> navbar header title-1 title-2 title-3
  paragraph button container row row-with-borders col-single
  col-half col-third col-fourth col-two-thirds picture
  round-picture unordered-list small-form form-label form-input
  form-input-large
2

```

3.16 pav. Naujai sukurto eskizų rinkinio žodynas

Nuskaičius informaciją iš eskizo ir atpažinus jame naudojamus žymėjimus, galima atlikti transformavimą į kitą pasirinktą aplinką, pavyzdžiui, į **.html* failą. Transformavimo informacija turi būti atitinkamai atnaujinta **-dsl-mapping.json* faile (**3.17 pav.**). Šiame faile naudojama žymėjimų informacija, kuri aprašyta žodyno faile (**3.16 pav.**) bei į koki kitą kuris žymėjimas yra transformuojamas.


```
1  "opening-tag": "{",
2  "closing-tag": "}",
3  "body": "<!doctype html>\n<html>\n<head>\n<meta charset=\"utf-8\">\n<meta name
4  "header": "",
5  "container": "<div class=\"container\">\n{\n}\n</div>\n",
6  "row-with-borders": "<div class=\"row\">\n{\n}\n</div>\n",
7  "row": "<div class=\"row\">\n{\n}\n</div>\n",
8  "col-single": "<div class=\"col-12\">\n{\n}\n</div>\n",
9  "col-half": "<div class=\"col-6\">\n{\n}\n</div>\n",
10 "col-third": "<div class=\"col-4\">\n{\n}\n</div>\n",
11 "col-fourth": "<div class=\"col-3\">\n{\n}\n</div>\n",
12 "col-two-thirds": "<div class=\"col-8\">\n{\n}\n</div>\n",
13 "navbar": "<nav class=\"navbar navbar-expand-lg navbar-light bg-light\">\n<div
14 "title-1": "<h1>[\n]\n</h1>\n",
15 "title-2": "<h2>[\n]\n</h2>\n",
16 "title-3": "<h3>[\n]\n</h3>\n",
17 "paragraph": "<p>[\n]\n</p>\n",
18 "button": "<a href=\"#\" class=\"btn btn-primary\">[\n]\n</a>\n",
19 "picture": "<img src=\"https://via.placeholder.com/1000\" alt=\"\" style=\"wid
20 "round-picture": "<img src=\"https://via.placeholder.com/1000\" alt=\"\" style
21 "unordered-list": "<ul>\n<li>[\n]\n</li>\n<li>[\n]\n</li>\n<li>[\n]\n</li>\n</ul>\n",
22 "small-form": "<div class=\"form-group\">\n<label>Input</label>\n<input type=
23 "form-label": "<label>[\n]\n</label>\n",
24 "form-input": "<input type=\"text\" placeholder=\"\">\n",
25 "form-input-large": "<textarea placeholder=\"\">\n</textarea>\n"
26
27
28
```

3.17 pav. Eskizo aprašo transformavimo į kitą formatą žodyno failas

3.2. Tinkamumo naudoti testavimas (angl. *usability testing*)

Tinkamumo naudoti testavimas skirtas norint išsiaiškinti ar sistema yra patogi naudotis potencialiems vartotojams bei ištestuoti bent dalį nefunkcinių reikalavimų.

Optimaliausia jeigu šis testavimas yra atliekamas su 3-7 potencialiais programinės įrangos naudotojais, su kiekvienu jų atliekant testavimą individualiai. Testavimo metu naudotojas atlieka užduotis sistemoje, pats suranda kaip jas atlikti, o tuo tarpu testuotojas stebi vartotoją ir užsirašinėja pastabas bei prižiūri testavimo eigą, pavyzdžiui, jei vartotojas yra per ilgai užstrigęs ties kažkuria užduotimi, tuomet padeda ar liepia pereiti prie kitos užduoties. Testavimo metu, su raštišku žmonių sutikimu galima daryti testavimo įrašą, kurį paskui galima parodyti komandos nariams ar testavimo eigą sudokumentuoti ir po testavimo, kad testavimo metu būtų galima daugiau dėmesio skirti stebėjimui. Testavimas su vienu vartotoju turėtų trukti apie 30-60 min., kad būtų galima pamatyti užtekinai bei tuo pačiu per daug nenuvarginti žmogaus.

Pasiruošimas testavimui

Atliekant tinkamumo naudoti testavimą buvo perskaitytos instrukcijos, pateiktos priede „5 priedas. Tinkamumo naudoti testavimo scenarijus“. Vaizdas ir garsas nebuvo įrašinėjamas, todėl gauti

sitikimo iš respondento nereikėjo. Testavimo metu, stebint kaip respondentas atlikinėja užduotis, viskas buvo užrašinėjama tekstu, tai pateikta priede „6 priedas. Tinkamumo naudoti testavimo užrašai“.

Kiekvieno respondento buvo paklausta šių asmeninių klausimų:

- Kuo šiuo metu užsiimi? Ką veiki per dienas?
- Kiek valandų per savaitę praleidi prie kompiuterio, įskaitant naršymą internete, įvairių programų naudojimą, darbą ar kita?
- Kokiuose tinklalapiuose paprastai lankaisi?
- Kokie tavo mėgstamiausi tinklalapiai?
- Ar naudojiesi sudėtingomis sistemomis, prie kurių būtina prisijungti? Pavyzdžiui, elektroninė bankininkystė.
- Kas tokiose sistemose tau labiausiai patinka?
- O kas labiausiai nepatinka?

Tuomet kiekvienam respondentui buvo užduodamos užduotys, arba kitaip įvardinus – veiklos, kurias turėjo išspręsti jie patys, be testuotojo (stebėtojo) pagalbos. Buvo stebima kaip respondentams sekėsi įgyvendinti šias užduotis (veiklas), pasižymint pagrindines mintis, bei tobulintinas vietas sistemoje. Testavimo veiklos buvo numatytos šios:

- Tikriausiai dar nieko nežinai apie sistemą. Ką darytum norėdama(-as) sužinoti daugiau informacijos (apie tai kas čia yra, kokios galimybės, bei kaip naudotis)?
- Pabandyk išsiaiškinti ir įvardinti kokią informaciją galima užpildyti apie kiekvieną projektą.
- Apsižvalgyk po projektą ir paaiškink ar supranti skirtumą tarp projektų, prototipų, langų, reikalavimų, užduočių ir kitų duomenų esybių?
- Šiai veiklai įsivaizduok, kad esi projekto vadovas(-ė). Atsirado naujas projektas apie kurį yra žinoma žemiau nurodyta informacija, kaip ją užpildytum?

Reikalavimai:

- Integruoti arba sukurti rezervacijų sistemą, kad būtų galima užsisakyti kambarį.
- Turi būti nuotraukų galerija, leidžianti pamatyti kiekvieną kambarį.
- Vartotojai turi galėti matyti savo profilį, aktyvias ar pasibaigusias rezervacijas, mokėjimus ir kitą susijusią informaciją.
- Būtina reprezentatyvi dalis, kurioje būtų informacija apie mus, kontaktai, kur randamės ir t.t.

Užsakovas:

- Jonas Jonauskas.

Prie projekto tinkamiausi dirbti darbuotojai:

- Gintarė Gintarytė.
- Tomas Tomauskas.
- Įsivaizduok, kad esi dizaineris(-ė), kuris projektuos visą vartotojo sąsają. Kokius prototipus bei langus šiame projekte susikurtum? Pabandyk juos sukurti. Jeigu prireiks, eskizų nuotraukos yra atidarytame aplanke. Kaip galėtum sugeneruoti ir peržiūrėti sistemos darbo rezultatus?
- Įsivaizduok, kad esi projekto vadovas(-ė). Pabandyk išsiaiškinti prie kurių projektų dabar dirba Tomas Tomauskas.
- Pabandyk išsiaiškinti kurie projektai buvo atšaukti.

- Pabandyk išsiaiškinti kaip sužinotum kurios užduotys iš bet kurio projekto vis dar nėra įgyvendintos.

Po testavimo dar buvo užduoti papildomi klausimai, kurie turėjo padėti įvertinti kai kurių nefunkcinių reikalavimų išpildymą. Kiekvieną klausimą buvo galima vertinti pagal Likerto skalę (angl. *Likert scale*), pažymint bet kurį variantą iš: visiškai nesutinku; nesutinku; nei sutinku, nei nesutinku; sutinku; visiškai sutinku; negaliu atsakyti. Buvo užduoti šie klausimai:

- Sistemoje esančių elementų dydis buvo tinkamas ir tai nebuvo sukėlę problemų bandant ant jų paspausti.
- Sistemoje dominavo minimalistinis ir šviesus stilius.
- Sistemos dizainas neblaškė dėmesio.
- Sistemos grafinė vartotojo sąsaja jautėsi neapkrauta.
- Svarbiausi veiksmai buvo lengvai pasiekiami.
- Sistemoje naudota terminologija buvo suprantama.
- Gauti prototipų langų generavimo rezultatai buvo gana tikslūs.

Į testavimą buvo pakviesti du respondentai. Viena iš jų potenciali sistemos naudotoja, grafikos dizainerė, kuriai kartas nuo karto tenka projektuoti grafines vartotojo sąsajas. Kitas – programuotojas, dažniausiai greitai susigaudantis bet kurioje sistemoje, tačiau nevisai susijęs su šia dalykine sritimi. Tokį testavimą geriausiai atlikti su 3-5 respondентаis, tačiau šiuo atveju, potencialių vartotojų, kurie testavimui skirtu laiku būtų pasiekiami – nebuvo.

Testavimo atlikimas

Testavimo metu buvo užsirašinėjami šie pagrindiniai dalykai:

- Pagrindinės mintys ką pasakė.
- Pagrindiniai veiksmai ką respondentas veikė.
- Kai kurios respondento emocinės reakcijos.
- Rastos sistemos klaidos.
- Pastebėjimai kas būtų patogiau vartotojui.
- Testuotojos (stebėtojos) komentarai.

Šie užrašai yra pateikti priede „6 priedas. Tinkamumo naudoti testavimo užrašai“.

3.3. Tinkamumo naudoti sistemą tyrimas

Tinkamumo naudoti įvertis (angl. *System Usability Scale (SUS)*)

Norint įvertinti sukurtos sistemos paprastumą, patikimumą ir kitus naudojimo aspektus, net ir su mažu naudotojų kiekiu, buvo pasirinkta panaudoti tinkamumo naudoti įvertį (angl. *System Usability Scale (SUS)*). Šį įvertį galima gauti sistemos naudotojams pateikiant klausimyną, kuris sudarytas iš tokių dešimties klausimų: [37]

1. Manau, kad norėčiau dažnai naudoti šį projektą (angl. *I think that I would like to use this project frequently*).
2. Man pasirodė, kad šis projektas yra be reikalo per sudėtingas (angl. *I found the project unnecessarily complex*).

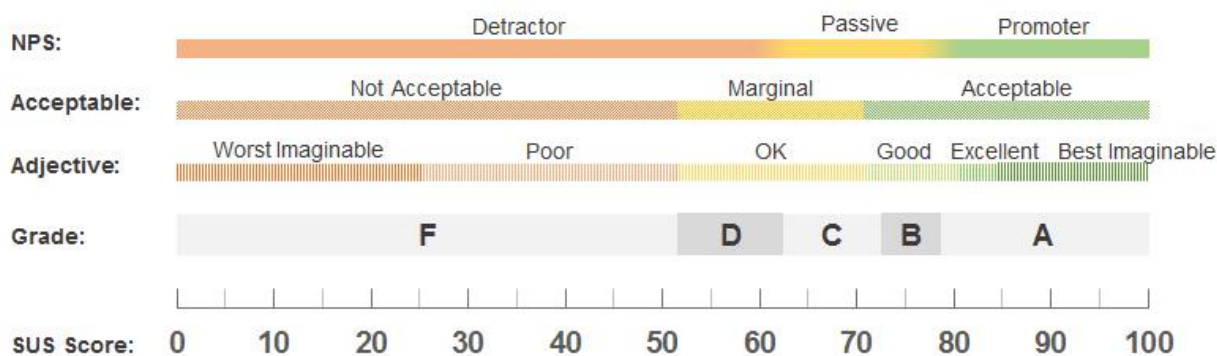
3. Man pasirodė, kad ši projektą lengva naudoti (angl. *I thought the project was easy to use*).
4. Manau, kad man prireiktų techninio asmens paramos, kad sugebėčiau naudotis šiuo projektu (angl. *I think that I would need the support of a technical person to be able to use this project*).
5. Supratau, kad įvairios šio projekto funkcijos buvo gerai integruotos (angl. *I found the various functions in this project were well integrated*).
6. Pagalvojau, kad šiame projekte yra per daug nenuoseklumo (angl. *I thought there was too much inconsistency in this project*).
7. Įsivaizduoju, kad žmonės galėtų labai greitai išmokti naudotis šiuo projektu (angl. *I imagine that most people would learn to use this project very quickly*).
8. Man pasirodė, kad projektą labai sunku naudoti (angl. *I found the project very cumbersome to use*).
9. Jaučiausi labai užtikrintai naudodamasis projektu (angl. *I felt very confident using the project*).
10. Man reikėjo išmokti daug dalykų, kad galėčiau susigaudyti šiame projekte ir pradėti naudoti (angl. *I needed to learn a lot of things before I could get going with this project*).

Sistemos naudotojai į šiuos klausimus atsakinėja pagal Likerto skalę (angl. *Likert scale*), kur į kiekvieną klausimą galima atsakyti: visiškai nesutinku; nesutinku; neutralu; sutinku; visiškai sutinku. Šie atsakymai išsiverčia į balus nuo 1 iki 5, kur 1 reiškia – „visiškai nesutinku“ ir 5 – „visiškai sutinku“.

Sistemos panaudojimo įverčio skaičiavimas

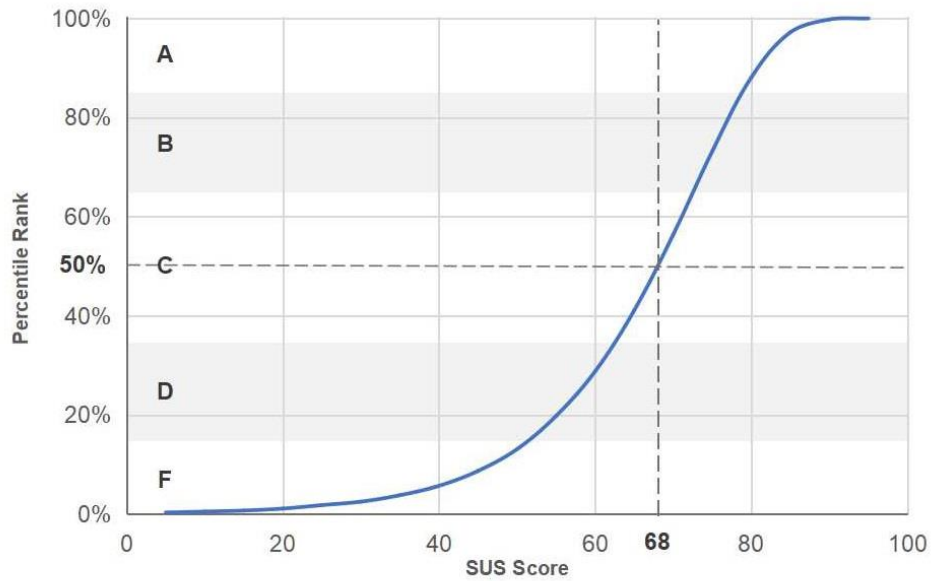
Šis įvertis skaičiuojamas pagal formulę, nurodytą metodologijoje [38, 39].

Tinkamumo naudoti įvertį (angl. *System Usability Scale (SUS)*) galima palyginti su kitokiomis matavimų sistemomis [39]. **3.18 pav.** matomas palyginimas tarp SUS, NPS, tinkamumo ir priėmimo.



3.18 pav. Įverčiai, tinkamumas naudoti, priėmimas ir NPS kategorijos, susijusios su SUS įverčiais [39]

Pagal procentus, vidutinė riba yra 50 proc., tačiau pagal SUS vidutinė riba yra 68 balai [39] (**3.19 pav.**). Tai reiškia, kad tai kas yra žemiau 68 balų yra žemiau vidurkio, o kas virš 68 balų – aukščiau už vidurkį.



3.19 pav. SUS grafike kartu su procentiniu įvertinimu ir balais [39]

Tinkamumo naudoti įvertinimo, tinkamumo, priimtimumo ir NPS palyginimas matomas **3.20 pav.**

Percentile range	Grading SUS Key		Acceptable
96 - 100	84.1-100	Best imaginable	Acceptable
65 - 96	72.6 - 84	Excellent	Acceptable
35 - 64	71.1-80.7	Good	Acceptable
15 - 34	51.7-71	OK/Fair	Marginal
2 - 14	25.1 – 51.6	Poor	Not Acceptable
0 - 1.9	0-25	Worst imaginable	Not Acceptable
	SUS	Grade	NPS
85 - 100	78.9-100	A	Promoter
65 - 84	72.6-78.8	B	Passive
35 - 64	62.7-72.5	C	Passive
15 - 34	51.7-62.6	D	Detractor
0 - 14	Less than 51.6	F	Detractor

3.20 pav. Tinkamumo naudoti įvertinimas (angl. *System Usability Scale (SUS)*), priimtimumas

4. Eksperimentinė dalis

4.1. Išplėsto modelio tyrimas

Tyrimo tikslas – nustatyti kokių tikslumu veikia modelis, kuris apmokytas įvedus įvairesnių stulpelių variacijų, paveikslukų, sąrašų, formų, kaip papildomus elementus.

Naudojama techninė ir programinė įranga

Kompiuterio ant kurio buvo generuojami duomenys ir vykdomas tyrimas parametrai:

- Procesorius – *Intel Core i3 7300 @ 4.00 GHz*.
- Operatyvioji atmintis (RAM) – *32 GB*.
- Vaizdo plokštė – *Nvidia GeForce GTX 1070 Ti (MSI)*.

Kompiuterio ant kurio buvo vykdomas modelio apmokymas ir vykdomas tyrimas parametrai:

- Procesorius – *AMD Ryzen 9 3950X 16-Core 3.49 GHz*.
- Operatyvioji atmintis (RAM) – *64 GB*.
- Vaizdo plokštė – *2080 Ti RTX*.

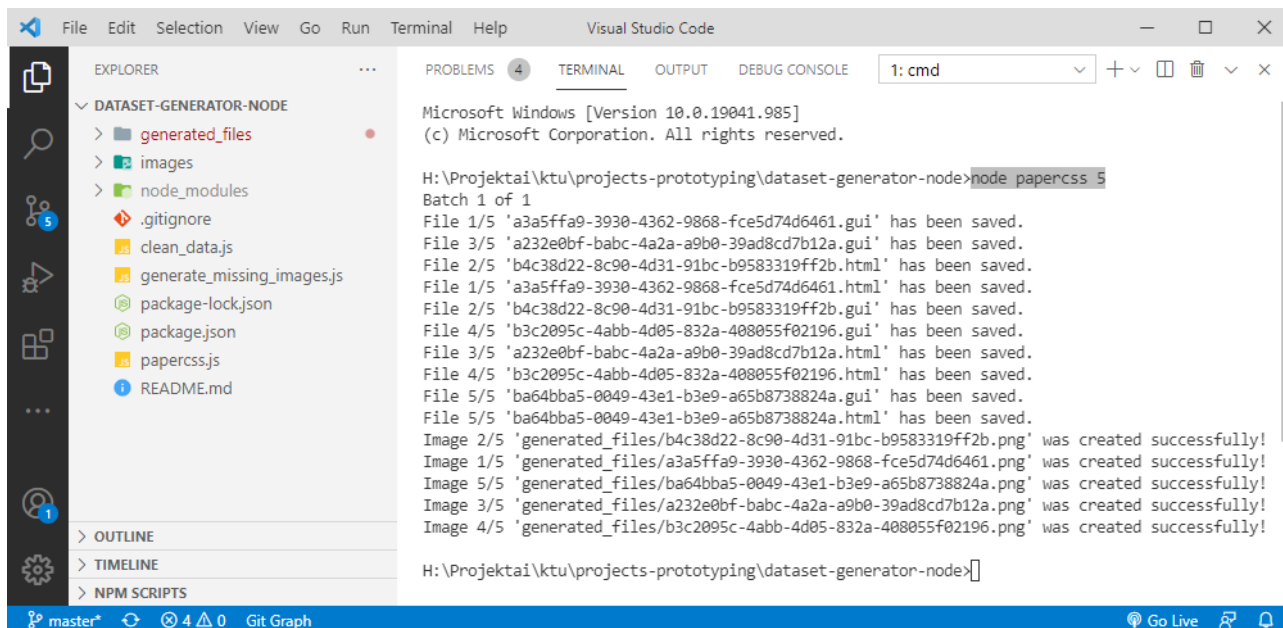
Tyrimas buvo vykdomas kaip pagrindą imant *pix2code* sprendimą, kurį galima pasiekti adresu: <https://github.com/tonybeltramelli/pix2code>.

Tyrimo eiga:

- Eskizų aprašų **.gui* formatu, **.html* failo ir eskizo paveiksluko **.png* formatu generavimas.
- Eskizų validavimas ir nevalidžių šalinimas.
- Duomenų rinkinio išskaidymas į 3 dalis (treniravimui, validavimui ir testavimui).
- Paveikslukų konvertavimas į *numpy* bibliotekos masyvus.
- Modelio apmokymas su 30 epochų, kur kiekviena epocha vykdoma su 72000 žingsnių.
- Modelio panaudojimas kodo generavimui pagal testavimui skirtą duomenų rinkinį.
- Originalaus ir sugeneruoto kodų palyginimas ir įvertinimas naudojant BLEU įvertį.

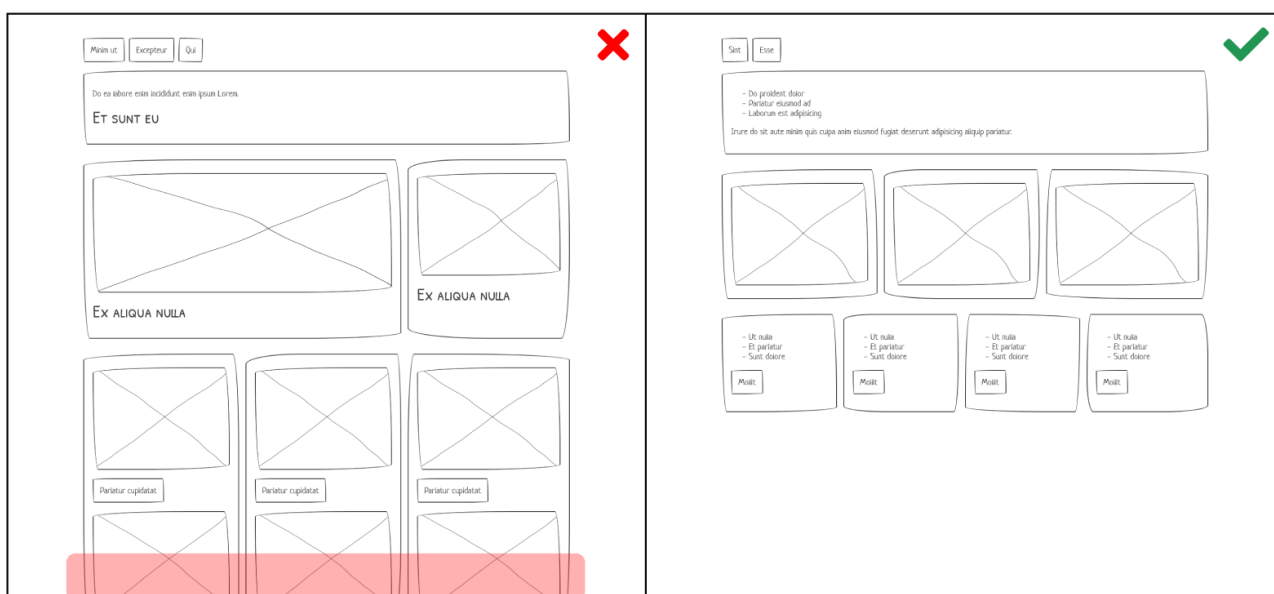
Norint pradėti eskizų generavimą, bus reikalingas *NodeJS* serveris, kuris galės įvykdyti generatoriuje aprašyta *JavaScript* programavimo kalba parašyta kodą. Esant generatoriaus kataloge, konsolėje yra naudojama komanda „*node papercss skaicius*“, kur „*skaicius*“ reiškia norimą sukurti eskizų kiekį, pavyzdžiui, „*node papercss 5*“ (**4.1 pav.**), neįrašius jokio skaičiaus, pagal numatytuosius parametrus bus generuojami 10 eskizų. Prasidėjus eskizų generavimui bus sukuriama po 5 eskizus vienu metu dėl tam tikrų operacinės sistemos ir naudojamų bibliotekų apribojimų, todėl generatorius automatiškai pasiskaičiuos kiek atskirų iteracijų bus vykdoma. Reikia turėti omenyje tai, kad paveikslukai yra sukuriami iš **.html* failų, o šis konvertavimas generavime užtrunka daugiausiai laiko, todėl sugeneruoti 2 tūkst. paveikslėlių gali užtrukti valandą.

Jeigu generavimo metu programa dėl kokių nors priežasčių bus nutraukta ir bus sukurti ne visi paveikslėliai iš turimų **.html* failų, trūkstamus paveikslėlius galima sukurti paleidus komandą „*node generate_missing_images*“.



4.1 pav. Eskizų generavimo paleidimas

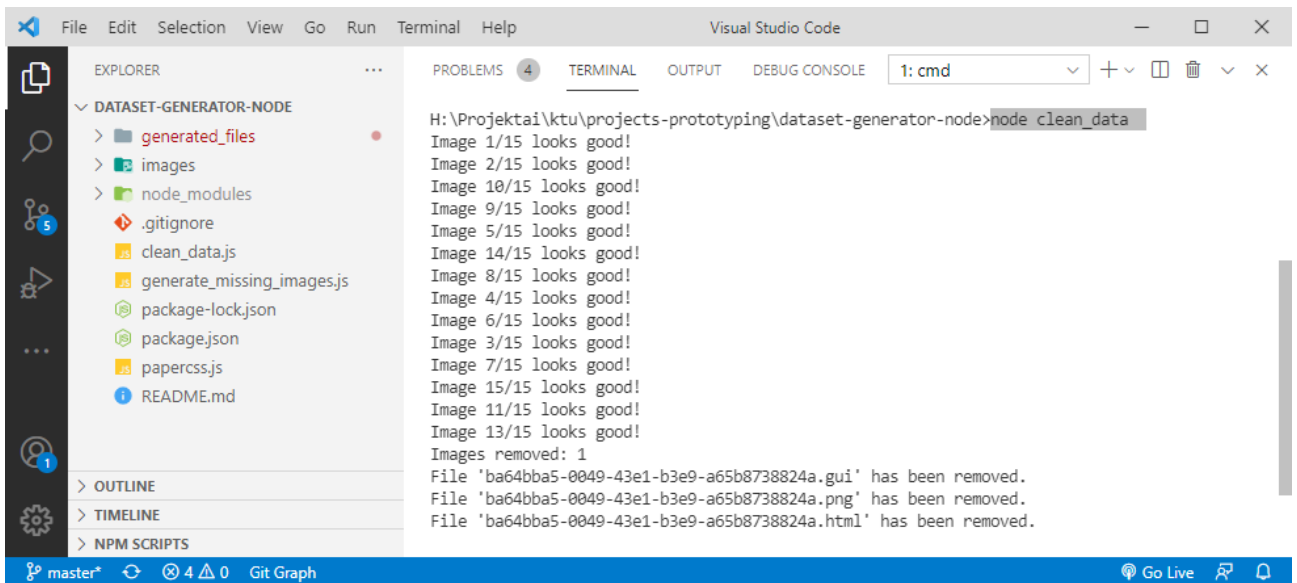
Dėl eskizų paveikslėlių rezoliucijos apribojimo bei dėl dinamiškai kuriamo eskizo turinio, kartais pasitaiko atveju, kai turinys netelpa į nuotraukos rėmus ir tam tikra dalis yra nukerpama. Eskizą galima laikyti tinkamu, kai jo turinys pilnai telpa į nuotraukos rėmus ir eskizą netinkamu galima laikyti kai jo turinys išlenda iš nuotraukos rėmų (4.2 pav.).



4.2 pav. Netinkamo eskizo dėl išlendančio turinio (kairėje) ir tinkamo eskizo (dešinėje) pavyzdžiai

Norint išvalyti netinkamus eskizus, tai galima atlikti rankiniu būdu peržiūrint eskizus ir rankiniu būdu ištriniant netinkamų eskizų *.png, *.html ir *.gui failus arba galima paleisti komandą „node clean_data“, kuri automatiškai pereis per visus rastus eskizus ir įsitikins, kad visuose juose turinys telpa į nuotraukos rėmus, o kitu atveju ištrintų eskizų ir su juo susijusius failus (4.3 pav.). Ši programa vienu metu peržiūri po 20 eskizų paveikslėlių ir iškart ištrina netinkamus. Kai nuotraukos yra tikrinamos, jų informacija yra įrašoma į operatyvią atmintį, todėl vienu metu tikrinant 20 eskizų yra

sunaudojama apie 1-2 GB operatyviosios atminties. Turint daugiau operatyviosios atminties, galima padidinti peržiūrimų eskizų kiekį, kad tikrinimas būtų įvykdytas greičiau.



4.3 pav. Eskizų validavimas ir netinkamų šalinimas

Duomenų rinkinio išskaidymas per *pix2code* į duomenis skirtus treniravimui ir validavimui atliekamas vykdant *build_datasets.py* failą, kuriam reikia nurodyti kur yra duomenys, kurie turi būti išskaidyti (4.4 pav.).

```
# split training set and evaluation set while ensuring no training example in the evaluation set
# usage: build_datasets.py <input path> <distribution (default: 6)>
./build_datasets.py ../datasets/ios/all_data
./build_datasets.py ../datasets/android/all_data
./build_datasets.py ../datasets/web/all_data
```

4.4 pav. Duomenų rinkinio išskaidymas į treniravimo ir validavimo rinkinius su *pix2code*

Kai duomenys išskaidyti į skirtingus rinkinius, juos toliau reikia paversti į *numpy* masyvus, o tai su *pix2code* įvyksta paleidžiant *convert_imgs_to_arrays.py* failą, kuriam yra perduodamas parengtas apmokymo rinkinys (4.5 pav.).

```
# transform images (normalized pixel values and resized pictures) in training dataset to numpy arrays
# usage: convert_imgs_to_arrays.py <input path> <output path>
./convert_imgs_to_arrays.py ../datasets/ios/training_set ../datasets/ios/training_features
./convert_imgs_to_arrays.py ../datasets/android/training_set ../datasets/android/training_features
./convert_imgs_to_arrays.py ../datasets/web/training_set ../datasets/web/training_features
```

4.5 pav. Eskizų nuotraukų pavertimas į *numpy* bibliotekos masyvus su *pix2code*

Modelio apmokymas per *pix2code* vyksta paleidus *train.py* failą ir nurodant treniravimui skirtą duomenų rinkinį bei kur turėtų būti išsaugotas apmokytas modelis (4.6 pav.).

```
# provide input path to training data and output path to save trained model and metadata
# usage: train.py <input path> <output path> <is memory intensive (default: 0)> <pretrained weights (optional)>
./train.py ../datasets/web/training_set ../bin
```

4.6 pav. Modelio apmokymas su *pix2code*

Kai modelis yra apmokytas, pabandyti sugeneruoti eskizų aprašus iš testavimui skirtų duomenų su *pix2code* galima paleidžiant *generate.py* failą (4.7 pav.). Paleidžiant šį failą reikia nurodyti turimą apmokytą modelį, eskizų nuotraukas bei lokaciją kur bus talpinami rezultatai.

```
# generate DSL code with beam search and a beam width of size 3
./generate.py ../bin pix2code ../gui_screenshots ../code 3
```

4.7 pav. Eskizų aprašų (*.gui) generavimas iš eskizų nuotraukų su *pix2code*

Taip pat, su *pix2code* galima generuoti vieno pasirinkto eskizo aprašymo failą iš nuotraukos paleidžiant *sample.py* failą (4.8 pav.).

```
# generate DSL code with beam search and a beam width of size 3
./sample.py ../bin pix2code ../test_gui.png ../code 3
```

4.8 pav. Eskizo aprašo (*.gui) generavimas iš eskizo nuotraukos su *pix2code*

BLEU įverčio skaičiavimo logiką galima gauti *SketchCode* projekte (4.9 pav.), kurį galima pasiekti adresu: <https://github.com/ashnkumar/sketch-code>. Norint paskaičiuoti tikslumą, gautą iš modelio prognozių, iš *SketchCode* galima paleisti *evaluate_single_gui.py* failą arba *evaluate_batch_guis.py* failą (4.9 pav.).

```
# evaluate single GUI prediction
python evaluate_single_gui.py --original_gui_filepath {path/to/original/gui/file} \
    --predicted_gui_filepath {path/to/predicted/gui/file}

# evaluate multiple GUI predictions
python evaluate_batch_guis.py --original_guis_filepath {path/to/folder/with/original/guis} \
    --predicted_guis_filepath {path/to/folder/with/predicted/guis}
```

4.9 pav. BLEU įverčio skaičiavimas su *SketchCode*

Tyrimo rezultatai

Modelis buvo apmokytas, eskizus papildant įvairesnėmis stulpelių variacijom, paveikslukais ir nerikiuotais sąrašais. Modelio apmokymui buvo panaudota 2100 eskizų paveikslėlių, apmokymas truko 30 epochų, kiekvienoje epochoje buvo vykdoma 72 tūkst. žingsnių. Tikslumas buvo įvertintas naudojant BLEU įvertį.

Iš 300 testavimui skirtų eskizų, bendras prognozių tikslumas siekia 61 proc., geriausias rezultatas 95 proc., o prasčiausias rezultatas – 29 proc. Geriausios prognozės rezultatas matomas 4.10 pav. (95

proc. pagal BLEU įvertį), šioje prognozėje vienintelis neatitikimas yra paskutinėje eilutėje, paskutiniame stulpelyje, kur trūksta paragrafo.

```
1 header {
2 button, button
3 }
4 row-with-borders {
5 col-fourth {
6 unordered-list, button
7 }
8 col-fourth {
9 unordered-list, button
10 }
11 col-fourth {
12 unordered-list, button
13 }
14 col-fourth {
15 unordered-list, button
16 }
17 }
18 row-with-borders {
19 col-third {
20 title-3
21 }
22 col-two-thirds {
23 title-3
24 }
25 }
26 row-with-borders {
27 col-single {
28 button, paragraph
29 }
30 }
31
```

```
1 header{
2 button,button
3 }
4 row-with-borders{
5 col-fourth{
6 unordered-list,button
7 }
8 col-fourth{
9 unordered-list,button
10 }
11 col-fourth{
12 unordered-list,button
13 }
14 col-fourth{
15 unordered-list,button
16 }
17 }
18 row-with-borders{
19 col-third{
20 title-3
21 }
22 col-two-thirds{
23 title-3
24 }
25 }
26 row-with-borders{
27 col-single{
28 button
29 }
30 }
31
```

4.10 pav. Originalus eskizo aprašas (kairėje) ir nuspėtas eskizo aprašas (dešinėje)

Į eskizus pavyko sėkmingai įterpti naujų stulpelių variacijų, paveikslėlių bei nerikiuotų sąrašų. Turint daugiau resursų apmokymą būtų galima paleisti su daugiau duomenų ir daugiau epochų, o tai turėtų ženkliai pagerinti gaunamus rezultatus.

4.2. Tinkamumo naudoti testavimas (angl. *usability testing*)

Tyrimo tikslas – rasti sistemos problemas, kurias gali trukdyti naudotojams ja naudoti ir rasti kitas tobulintinas sistemos dalis, kurios sutvarkius naudotojams turėtų būti patogiau naudotis sistema.

Naudojama techninė ir programinė įranga

Kompiuterio ant kurio buvo patalpintas projektas ir vykdomas tyrimas parametrai:

- Procesorius – *Intel Core i3 7300 @ 4.00 GHz.*
- Operatyvioji atmintis (RAM) – *32 GB.*
- Vaizdo plokštė – *Nvidia GeForce GTX 1070 Ti (MSI).*

Sistemą testavę respondentai jungėsi iš savo kompiuterių per *Team Viewer* priemonę ir valdė kompiuterį, kuriame patalpintas projektas.

Testavimo užrašai buvo rašomi per atskirą kompiuterį nuo to, kuriame buvo vykdomas tyrimas.

Tyrimo eiga:

- Respondentas supažindinamas su testavimu ir eiga, perskaitoma instrukcija.
- Respondentui perskaitoma po vieną užduotį / veiklą, kurią jis turi atlikti sistemoje. Dirbama iki kol baigiasi laikas arba nebelieka užduočių.
- Kol respondentas atlieka užduotį, visi pastebėjimai yra užrašinėjami.
- Pasibaigus laikui arba užduotims yra užduodami keli papildomi klausimai.
- Praeiti žingsniai kartojami su kiekvienu kitu respondentu.
- Aprašomas pastebėjimų apibendrinimas ir iškeliamos spęstinės vietos.

Tyrimo rezultatai

Atlikus tinkamumo naudoti testavimą, tobulintinos sistemos vietos buvo apibendrintos, o rezultatai pateikti **4.1 lentelėje** ir **4.2 lentelėje**.

Surastos sistemos klaidos matomos **4.1 lentelėje**. Šioje lentelėje apibūdintos sistemos klaidos, kurios gali sutrikdyti naudotojų darbą, yra pateikti aprašymai kaip šias klaidas galima atkartoti bei koks tokios klaidos sutvarkymo prioritetas.

4.1 lentelė. Tinkamumo naudoti testavimo metu surastos sistemos klaidos

Sistemos klaida	Kaip klaida įvyko	Prioritetas
Projekto kūrimas iš viršutinės juostos, sukūrus projektą nukreipia ne į tą projektą.	Peržiūrint projektą buvo paspausta ant „Naujas projektas“ mygtuko esančio viršutinėje juostoje. Supildžius informaciją ir paspaudus ant mygtuko „Sukurti“ projektas turėtų susikurti ir vartotojas turėtų būti nukreipiamas į ką tik sukurtą projektą. Projektas buvo sukurtas sėkmingai, tačiau vartotojui buvo parodytas ne tas projektas, dėl ko naujo projekto reikėjo ieškoti per sistemą.	Aukštas.
Neįkėlus eskizo į prototipo langą, prototipo generavimo mygtukas gali sutrikdyti sistemos darbą.	Vartotojas kuria prototipo langą, suveda visą reikiamą informaciją, tačiau neįkelia eskizo. Tuomet paspaudžia ant mygtuko „Sukurti“. Sistema sukuria naują langą, nukreipia vartotoją į jo peržiūrą. Šiame lange esančioje skiltyje „Prototipas“, vartotojas gali matyti mygtuką „Generuoti“, tačiau neįkėlus eskizo šis mygtukas bus be teksto, bei matysis tik mygtuko kontūrai. Vartotojui paspaudus ant tokio mygtuko, sistemos API dalis išmeta kritinę klaidą.	Aukštas.

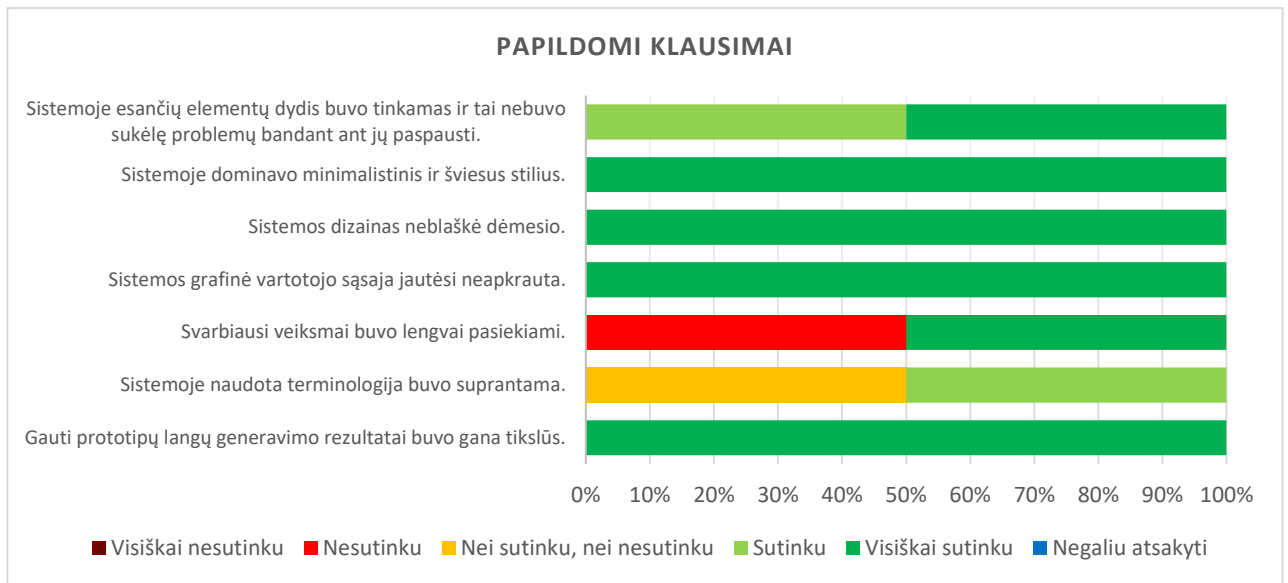
Testavimo metu ir po testavimo apgalvoti patobulinimai sistemoje nurodyti **4.2 lentelėje**. Šioje lentelėje nurodytos tobulintinos sistemos vietos, kurias sutvarkius sistemos naudotojams būtų patogiau atlikti vieną ar kitą užduotį, susirasti reikiamą informaciją ar atlikti kitus veiksmus sistemoje. Kiekvienas nurodytas tobulinimas turi aprašymą kaip jis gali būti realizuotas ar kas turėtų būti padaryta, yra nurodymas tokio pasiūlymo pagrindimas ar priežastis bei nurodytas tokio patobulinimo svarbumas aprašant jo prioritetą.

4.2 lentelė. Dėl tinkamumo naudoti testavimo rezultatų siūlomos tobulintinos vietos

Siūlomas tobulinimas	Pagrindimas	Prioritetas
Dokumentacijos pradiniam puslapyje įdėti trumpą informaciją apie sistemą, duomenis joje, galimybes. Informacija turėtų būti trumpesnė už „Greita pradžia“.	Abu respondentai tikėjosi rasti kokį nors paragrafą apie pačią sistemą, kokie joje yra duomenys ir kokios sistemos galimybės. Abu respondentai šios informacijos ieškojo ten pat.	Aukštas.
Dokumentacijoje pervadinti nuorodą „Greita pradžia“ į „Pirmas naudojimasis sistema“.	Abu respondentai susipainiojo kokią informaciją ras paspaudę ant nuorodos „Greita pradžia“, tačiau ją panagrinėjus suprato, kad ten pirmo naudojimo instrukcijos.	Aukštas.
Dokumentacijoje pervadinti nuorodą „Naudotojo atmintinė“ į „Pilna dokumentacija“.	Abu respondentai nesuprato, kad paspaudę ant nuorodos „Naudotojo atmintinė“ ras pilną sistemos dokumentaciją. Tik atsitiktinumo dėka paspaudė ir panagrinėjo ten esančią informaciją, kas jiems pasirodė labai naudinga dėl išsamumo.	Aukštas.
Įdėti interaktyvų supažindinimo gidą, kuris vartotoją su jo sutikimu pravestų pro sistemą ir supažindintų su joje esančiais duomenimis bei jos galimybėmis. Arba įdėti pasirodančių patarimų (angl. <i>tooltips</i>) su papildoma informacija ne vien prie kūrimo formų, pavyzdžiui, prie kiekvieno lango ar bloko antraščių.	Abiem respondentams buvo šiek tiek sudėtinga pradžia, nes nesuprato skirtumo tarp projektų, prototipų ir langų. Taip pat, reikėjo ieškoti informacijos apie tai ką dar galima padaryti. Tai galėtų išspręsti interaktyvus gidas, kuris parodytų kur kas yra ir ką galima atlikti. Arba bent jau būtų galima padubliuoti informacijos iš dokumentacijos ir ją pridėti iškart prie įvairių langų ar blokų antraščių.	Aukštas.
Projekte sukuriant prototipą, reikalavimą ar užduotį, bei prototipe sukuriant langą, vartotojas turėtų būti paliktas tame pačiame lange ir nenukreipiamas į sukurtos esybės informaciją.	Vienas iš respondentų išreiškė norą likti tame pačiame lange sukūrus kokių nors duomenų. O kitas respondentas pasikloviė sistema ir atsidarius daugiau informacijos pasitikrindavo ar viskas susikūrė tvarkingai, tačiau lyg ir norėjo likti tame pačiame lange iš kurio kūrė naujus duomenis.	Vidutinis.
Pridėti paaiškinimų sistemoje apie prototipą ir prototipų langus, panaudojant pavyzdžius. Gal tekstu kur nors netoliесе.	Abu respondentai labai maišėsi tarp prototipų ir langų, bei kokią reikšmę jie turi. Informacijos papildomai ieškojo dokumentacijoje, tačiau matėsi noras suprasti esant pačioje sistemoje.	Vidutinis.
Prie eskizų pridėti paaiškinimą kur ir kaip jie įkeliami.	Vienas respondentas, net vykdydamas kitas veiklas, pastebėjo eskizus ir jam iškart kilo klausimų kur, kada ir kaip jie yra įkeliami, nes nerado eskizų įkėlimo šalia jų.	Vidutinis.
Projekto kūrimo formoje pridėti informacijos apie tai kada bus kuriami prototipai, reikalavimai, užduotys, bei priskiriami vartotojai.	Vienas respondentas ne iškart suprato, kad su projektu susijusių duomenų kūrimas yra leidžiamas tik sukūrus projektą.	Vidutinis.
Prototipo lango skiltyje „Prototipas“ pridėti informacijos apie ten esančius mygtukus.	Abu respondentai nesuprato ko gali tikėtis nurodytoje vietoje. Ne iš karto suprato kokius rezultatus gaus, bei ką jie reiškia.	Vidutinis.
Leisti generuoti (apdoroti) kelis prototipo langus.	Vienas respondentas ieškojo kur galėtų atlikti tokį veiksmą su keliais langais vienu metu. Įvardino, kad jeigu turėtų sukelti daug langų, tuomet tą patį turėtų atlikti prie kiekvieno atskirai.	Vidutinis.

Prototipo lango informacijos puslapyje, kai jau yra sugeneruoti rezultatai, mygtuką „Generuoti“ pervadinti į „Generuoti iš naujo“.	Abu respondentai, sugeneravę rezultatus, vis dar matydavo mygtuką „Generuoti“, kuris juos sumaišydavo, nes nebuvo aišku ką tas mygtukas atlieka – ar atlieka tą patį ką ir anksčiau, ar pasikeitė šio mygtuko prasmė, ar jį rodo kai neturėtų rodyti?	Vidutinis.
Projekto informacijoje esančią skiltį „Susiję vartotojai“ pervadinti arba prie jos pridėti daugiau informacijos apie tai kas yra šioje skiltyje, kad būtų galima suprasti nespaudžiant ant joje esančio mygtuko „Priskirti naują vartotoją“.	Vienas iš respondentų iš pradžių susimaišė ar užsakovą bei darbuotojus galima pridėti iš susijusių vartotojų skilties ar tai turėtų vykti kažkur kitur, pavyzdžiui, projekto pagrindinėje informacijoje. Nors ir veiklą įgyvendino sėkmingai, tačiau tik dėl to, kad po dvejones pabandė paspausti ant mygtuko „Priskirti naują vartotoją“ ir ten rado tai ko ieškojo.	Žemas.
Sugalvoti naują terminologiją duomenų esybėms: prototipai; prototipų langai.	Respondentai per ilgai užtruko suprasti kokią prasmę turi šios duomenų esybės bei kuo jos skiriasi. Vienas iš respondentų įvardino, kad norėjo jog žodis „Prototipai“ būtų užvadintas kaip nors kitaip.	Žemas.
Suspausti reikalavimų skiltyje esančią informaciją, kuri pateikiama sąrašo pavidalu.	Vienas iš respondentų įvardino, jog norėtų, kad reikalavimų skiltyje, kai duomenys pateikiami sąrašo pavidalu, būtų kaip nors sutraukti, nes jo nuomone yra išnaudojama per daug vietos ir reikia daug vaikščioti su pele.	Žemas.
Prototipo lango skiltyje „Pasiūlymai“ pridėti informacijos apie tai kas ten yra.	Abiem respondentams nurodytoje vietoje buvo neaiškumų, nei vienas nesuprato kas ten yra ir ko tikėtis.	Žemas.
Pridėti duomenų rūšiavimo filtrą.	Prie tam tikrų testavimo veiklų abu respondentai ieškojo duomenų filtravimo to konkretaus bloko viršuje, netoliese duomenų peržiūros būdo pasirinkimo ir paieškos. To neradę, pasirinkdavo peržiūrėti duomenis lentelės pavidale ir rikiuodavo duomenis pagal norimą stulpelį. Nors ir veiklos įgyvendintos sėkmingai, tačiau rikiavimas pagal stulpelį nebūtinai visada išspręs visus paieškos norus.	Žemas.
Padaryti, kad iš dokumentacijos būtų galima grįžti į sistemą.	Dokumentacija atsidaro naujame naršyklės skirtuke, tačiau abu respondentai išreiškė norą, kad į sistemą būtų galima grįžti ir paspaudžiant nuorodą dokumentacijoje, o ne uždarant skirtuką.	Žemas.
Prie užduočių (ne užduoties informaciniame puslapyje), sutvarkyti informaciją kuri atvaizduojama. Panaikinti informaciją, kuri mažiausiai aktuali, kad nebūtų per daug informacijos pateikiama vartotojui ir jo neblaškytų. Pridėti projektą, kad matytųsi kuriame projekte yra užduotis, išskyrus tą atvejį kai užduotys yra atvaizduojamos pačiame projekte.	Vienas respondentas išsigando užduočių puslapyje pateikiamos informacijos kiekiu ir tuo pačiu nerado kaip galėtų užduotis susirikiuoti pagal projektus ar pamatyti kuriam projekte kuri užduotis yra sukurta.	Žemas.

Testavimo pabaigoje papildomai užduotų klausimų rezultatai matomi **4.11 pav.** Šie klausimai buvo užduoti norint išsiaiškinti papildomos informacijos apie naudotojų savijautą po sistemos testavimo bei kaip jie vertina tam tikrus dalykus, kurie susiję su nefunkciniais sistemos reikalavimais.



4.11 pav. Tinkamumo naudoti testavimo papildomų klausimų rezultatai

4.3. Tinkamumo naudoti sistemą tyrimas

Tyrimo tikslas – išsiaiškinti koks yra sistemos tinkamumas naudoti pagal SUS skalę.

Naudojama techninė ir programinė įranga

Kompiuterio ant kurio buvo patalpintas projektas ir vykdomas tyrimas parametrai:

- Procesorius – *Intel Core i3 7300 @ 4.00 GHz.*
- Operatyvioji atmintis (RAM) – *32 GB.*
- Vaizdo plokštė – *Nvidia GeForce GTX 1070 Ti (MSI).*

Sistemą testavę respondentai jungėsi iš savo kompiuterių per interneto naršyklę.

Tyrimo eiga:

- Respondentų supažindinimas su tyrimo eiga.
- Respondentų supažindinimas su tiriamą programa.
- Respondentų naudojimas programa.
- Respondentų apklausa.

Sistemos testavimas ar naudojimas, kurį atliko naudotojai savo nuožiūra ir savo laiku. Tai reiškia, kad naudotojai galėjo ir turėjo su sistema susipažinti patys, lyg tai būtų produktas skirtas jiems, o tokiu atveju dažniausiai nėra pagalbinio žmogaus, kuris paaiškintų kaip naudotis sistema. Bandyką pasinaudoti sistemą naudotojai galėjo atlikti interneto naršyklės pagalba, prisijungdami per sistemos iš savo kompiuterio.

Klausimyno užpildymas, kurio metu naudotojas turėjo įvertinti 10 klausimų pagal Likerto skalę, t. y. kiekvieną klausimą pažymint kaip „visiškai nesutinku“, „nesutinku“, „neutralu“, „sutinku“, „visiškai sutinku“. Šį klausimyną užpildė 11 sistemos naudotojų.

Rezultatų apdorojimas, tinkamumo naudoti įverčio (angl. *System Usability Scale (SUS)*) paskaičiavimas ir rezultatų apibendrinimas.

Tyrimo rezultatai

Sistemos naudotojų atsakymai į klausimyną pateikti **4.3 lentelėje**. Šiame klausimyne buvo 10 klausimų, kuriuos naudotojai turėjo atsakyti. Šį klausimyną užpildė 11 naudotojų. Šioje lentelėje 1 reiškia – „visiškai nesutinku“, 2 – „nesutinku“, 3 – „neutralu“, 4 – „sutinku“, 5 – „visiškai sutinku“.

4.3 lentelė. Tinkamumo naudoti klausimyno rezultatai

Dalyvis(-ė) nr.	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
1	3	2	4	1	3	2	4	1	4	1
2	4	2	4	1	4	1	5	2	2	2
3	4	1	4	3	4	1	5	1	2	2
4	5	1	5	1	5	2	5	1	4	2
5	3	2	3	2	4	3	3	2	3	2
6	4	2	4	2	4	2	5	1	3	1
7	3	1	4	1	5	1	5	1	4	1
8	4	2	5	1	5	1	4	1	4	1
9	4	1	5	1	5	1	5	1	4	1
10	4	1	4	1	3	3	5	2	4	1
11	2	4	1	3	3	5	3	4	1	3

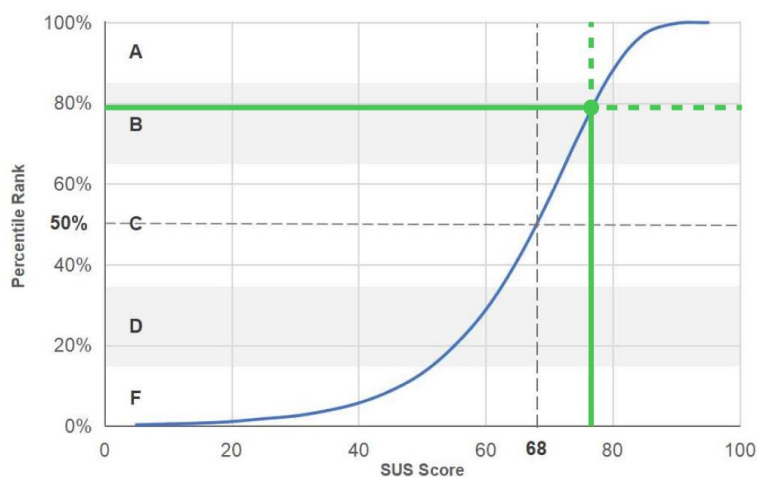
Pagal šiuos klausimyno rezultatus (**4.3 lentelė**) buvo paskaičiuotas sistemos tinkamumo naudoti įvertis (angl. *System Usability Scale (SUS)*), rezultatai pateikti **4.4 lentelėje**. Naudotojai sistemą įvertino 77,27 balams, kas reiškia, kad tinkamumas naudotis sistema yra priimtinas, tačiau dar yra kur tobulėti.

4.4 lentelė. Tinkamumo naudoti įverčio skaičiavimai

Dalyvis(-ė)	Įvertis							
	Nelyginiai kl.	Lyginiai kl.	SUS įvertis (/100)	Pažymys	Tinkamumas	NPS, %	NPS	Priimtinas
1	13	18	77,5	B	Gerai	100	Pasyvus	Priimtina
2	14	17	77,5	B	Gerai	16	Pasyvus	Priimtina
3	14	17	77,5	B	Gerai	16	Pasyvus	Priimtina
4	19	18	92,5	A	Puikiai	100	Pakylėjantis	Priimtina
5	11	14	62,5	D	Vidutiniškai	100	Tempiantis žemyn	Svertinis
6	15	17	80	A	Gerai	2	Pakylėjantis	Priimtina
7	16	20	90	A	Puikiai	51	Pakylėjantis	Priimtina

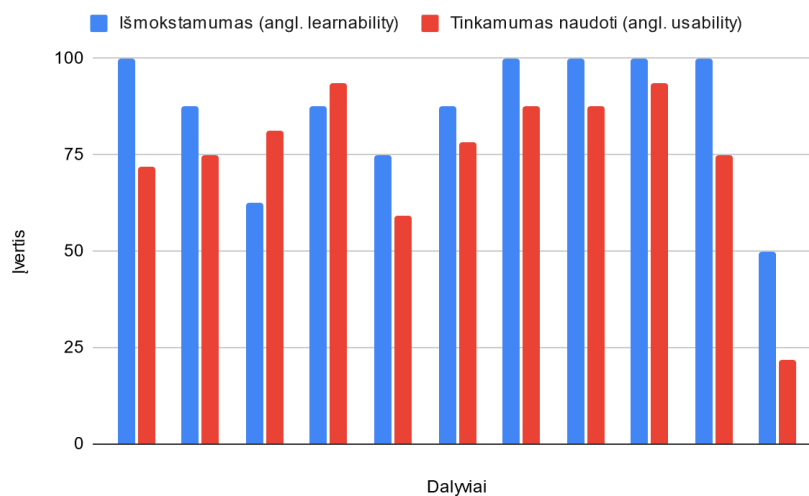
8	17	19	90	A	Puikiai	100	Pakylėjantis	Priimtina	
9	18	20	95	A	Puikiai	94	Pakylėjantis	Priimtina	
10	15	17	80	A	Gerai	2	Pakylėjantis	Priimtina	
11	5	6	27,5	F	Silpnai	44	Tempiantis žemyn	Nepriimtina	
			Vidutinis įvertis	77,27	B	Gerai	72	Pasyvus	Priimtina

Sistemos tinkamumo naudoti įvertis kartu su procentiniu įverčiu matomas **4.12 pav.** Sistemos tinkamumas naudoti buvo įvertintas 77,27 balo, tai atitinka maždaug 68 proc.



4.12 pav. Sistemos tinkamumo naudoti įvertis kartu su procentiniu įverčiu

Išmokstamumo (angl. *learnability*) ir tinkamumo naudoti (angl. *usability*) įverčiai pagal dalyvius matomi **4.13 pav.**. Šiame grafike galima pastebėti, kad išmokstamumas naudotis sistema 5 kartus siekia 100 balų, o kitais atvejais, išskyrus du atvejus, yra gana aukštas – 75 ar daugiau balų. Tuo tarpu tinkamumas naudoti yra įvertintas šiek tiek mažesniais balais beveik visais atvejais, tačiau yra įvertinta vis tiek sąlyginai gerai.



4.13 pav. Išmokstamumo (angl. *learnability*) ir tinkamumo naudoti (angl. *usability*) įverčiai pagal dalyvius

Išvados

1. Reikalavimus programinei įrangai buvo nuspręsta išgauti nagrinėjant prototipus (panašias sistemas). Tai leido išgryninti sistemos reikalavimus, kuriai sudaryta specifikacija.
2. Išanalizavus egzistuojančius dirbtinio intelekto sprendimus skirtus prototipų generavimui nuspręsta, kad kuriamos sistemos realizacijai labiausiai tiks *pix2code* ir *SketchCode* sprendimai, dėl sprendimo prieinamumo ir plečiamumo galimybių.
3. Atsižvelgus į funkcinis ir nefunkcinis reikalavimus buvo pasirinktas į paslaugas orientuotas architektūrinis stilius (SOA), kuris leis plėsti programinę įrangą naudojant įvairias technologijas.
4. Saityno paslaugų realizacija buvo pasirinkta *ASP.NET Core* architektūra. Tai leidžia įgyvendinti reikalavimus dėl sistemos plečiamumo, pernešamumo ir pritaikymo kitoms platformoms.
5. Kliento programos realizacijai buvo pasirinkta *Angular* karkaso architektūra. Tai leis ateityje įgyvendinti reikalavimus susijusius su sistemos pritaikymu kitoms platformoms.
6. Sistemos tinkamumo naudoti testavimas buvo atliktas su 2 respondentais. Testavimo metu išsakytos respondentų pastabos leido pasiūlyti 18 sprendimų, kurie leistų patobulinti sistemą, kad naudotojams būtų patogiau ja naudotis.
7. Norint įvertinti sistemos tinkamumą naudoti, buvo pasinaudota tinkamumo naudoti įverčiu (SUS). Paaiškėjo, kad naudotojai sistemą vertina gana gerai – 77,27 balo pagal SUS įvertį. Šis įvertis rodo, kad sistemos tinkamumas naudoti yra didesnis nei vidutinis (68 proc.) ir įvertinimas yra „gerai“.
8. Siekiant sukurti sprendimą, kuris būtų arčiau realaus pasaulio sąlygų, į eskizus buvo įtraukta papildomų žymėjimų, tokių kaip paveikslukai, nerikiuoti sąrašai, įvairesnės stulpelių variacijos ir t. t.. Modelio apmokymui buvo naudojamas 2100 eskizų (duomenų) rinkinys, apmokymą sudarė 30 epochų. Modelio tikslumas įvertintas pagal BLEU įvertį, bendras tikslumas siekia 61 proc., geriausio rezultato tikslumas – 95 proc., blogiausio rezultato tikslumas – 29 proc. Norint pagerinti šiuos rezultatus būtų galima modelį mokyti su daugiau duomenų bei apmokymo trukmei parinkti didesnę epochų skaičių.

Literatūros sąrašas

1. WALKER, M., TAKAYAMA, L. and LANDAY, J.A. *High-Fidelity Or Low-Fidelity, Paper Or Computer? Choosing Attributes when Testing Web Prototypes*. SAGE Publications Sage CA: Los Angeles, CA, 2002.
2. LIM, Y., PANGAM, A., PERIYASAMI, S. and ANEJA, S. *Comparative Analysis of High-and Low-Fidelity Prototypes for More Valid Usability Evaluations of Mobile Devices*. ACM, 2006.
3. SEFELIN, R., TSCHELIGI, M. and GILLER, V. *Paper Prototyping-what is it Good for? A Comparison of Paper-and Computer-Based Low-Fidelity Prototyping*. , 2003.
4. BUTLER, T. and FITZGERALD, B. A Case Study of User Participation in the Information Systems Development Process. *ICIS 1997 Proceedings*, 1997. pp. 27.
5. SAUER, J. and SONDEREGGER, A. The Influence of Prototype Fidelity and Aesthetics of Design in Usability Tests: Effects on User Behaviour, Subjective Evaluation and Emotion. *Applied Ergonomics*, 2009, vol. 40, no. 4. pp. 670-677.
6. VIRZI, R.A., SOKOLOV, J.L. and KARIS, D. *Usability Problem Identification using both Low-and High-Fidelity Prototypes*. ACM, 1996.
7. GRADY, H.M. *Web Site Design: A Case Study in Usability Testing using Paper Prototypes*. IEEE Educational Activities Department, 2000.
8. GERBER, E. and CARROLL, M. The Psychological Experience of Prototyping. *Design Studies*, 2012, vol. 33, no. 1. pp. 64-84.
9. SNYDER, C. *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann, 2003.
10. DUMAS, J.S., DUMAS, J.S. ir REDISH, J. *A practical guide to usability testing*. Intellect books, 1999.
11. RUDD, J., STERN, K. and ISENSEE, S. Low Vs. High-Fidelity Prototyping Debate. *Interactions*, 1996, vol. 3, no. 1. pp. 76-85.
12. COYETTE, A., KIEFFER, S. and VANDERDONCKT, J. *Multi-Fidelity Prototyping of User Interfaces*. Springer, 2007.
13. BLUMENSTEIN, M., CHENG, C.K. and LIU, X.Y. *New Preprocessing Techniques for Handwritten Word Recognition*. , 2002.
14. ESPANA-BOQUERA, S., CASTRO-BLEDA, M.J., GORBE-MOYA, J. and ZAMORA-MARTINEZ, F. Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, vol. 33, no. 4. pp. 767-779.
15. EL ABED, H. and MARGNER, V. *Comparison of Different Preprocessing and Feature Extraction Methods for Offline Recognition of Handwritten Arabicwords*. IEEE, 2007.

16. CAO, H. and GOVINDARAJU, V. *Handwritten Carbon Form Preprocessing Based on Markov Random Field*. IEEE, 2007.
17. BIENIECKI, W., GRABOWSKI, S. and ROZENBERG, W. *Image Preprocessing for Improving Ocr Accuracy*. IEEE, 2007.
18. CAO, H. and GOVINDARAJU, V. Preprocessing of Low-Quality Handwritten Documents using Markov Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, vol. 31, no. 7. pp. 1184-1194.
19. BOW, S.T. *Pattern recognition and image preprocessing*. CRC press, 2002.
20. HSE, H.H. and NEWTON, A.R. Recognition and Beautification of Multi-Stroke Symbols in Digital Ink. *Computers & Graphics*, 2005, vol. 29, no. 4. pp. 533-546.
21. HUANG, B.Q., ZHANG, Y. and KECHADI, M.T. *Preprocessing Techniques for Online Handwriting Recognition*. IEEE, 2007.
22. SEZGIN, T.M., STAHOVICH, T. and DAVIS, R. ACM SIGGRAPH 2007 courses, 2007 *Sketch Based Interfaces: Early Processing for Sketch Understanding*, pp. 37-es.
23. IGARASHI, T., MATSUOKA, S., KAWACHIYA, S. and TANAKA, H. ACM SIGGRAPH 2007 courses, 2007 *Interactive Beautification: A Technique for Rapid Geometric Design*, pp. 18-es.
24. ALVARADO, C.J. A Natural Sketching Environment: Bringing the Computer into Early Stages of Mechanical Design. *A Natural Sketching Environment: Bringing the Computer into Early Stages of Mechanical Design*, 2000.
25. ROBINSON, A. Sketch2code: Generating a Website from a Paper Mockup. *arXiv Preprint arXiv:1905.13750*, 2019.
26. SOBEL, I. An Isotropic 3x3 Image Gradient Operator, Presentation at Stanford AI Project 1968, 2014.
27. CANNY, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, no. 6. pp. 679-698.
28. LIU, Y., et al. *Richer Convolutional Features for Edge Detection*. , 2017.
29. LECUN, Y., TOURESKY, D., HINTON, G. and SEJNOWSKI, T. *A Theoretical Framework for Back-Propagation*. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.
30. HEARST, M.A., et al. Support Vector Machines. *IEEE Intelligent Systems and their Applications*, 1998, vol. 13, no. 4. pp. 18-28.
31. BREIMAN, L. Random Forests. *Machine Learning*, 2001, vol. 45, no. 1. pp. 5-32.
32. EVERINGHAM, M., et al. The Pascal Visual Object Classes (Voc) Challenge. *International Journal of Computer Vision*, 2010, vol. 88, no. 2. pp. 303-338.
33. BELTRAMELLI, T. *Pix2code: Generating Code from a Graphical User Interface Screenshot*. , 2018.

34. BALOG, M., et al. Deepcoder: Learning to Write Programs. *arXiv Preprint arXiv:1611.01989*, 2016.
35. GAUNT, A.L., et al. Terpret: A Probabilistic Programming Language for Program Induction. *arXiv Preprint arXiv:1608.04428*, 2016.
36. LING, W., et al. Latent Predictor Networks for Code Generation. *arXiv Preprint arXiv:1603.06744*, 2016.
37. BROOKE, J. Sus: A “quick and Dirty” usability. *Usability Evaluation in Industry*, 1996, vol. 189.
38. LEWIS, J.R. and SAURO, J. Item Benchmarks for the System Usability Scale. *Journal of Usability Studies*, 2018, vol. 13, no. 3.
39. J. SAURO. *5 ways to interpret a SUS score* [interaktyvus]. 5 [žiūrėta 2021-05-20]. Prieiga per: <https://measuringu.com/interpret-sus-score/>.

Priedai

1 priedas. Duomenų modelio specifikacijos

Duomenų modelio „Asmuo“ klasės ir jos atributų aprašymas

Duomenų žodyno elemento pavadinimas	Turinys	Tipas
Asmuo	Įmonės klientas (užsakovas) arba įmonės darbuotojas.	Klasė
Vardas	Asmens vardas. Privaloma nurodyti.	Atributas
Pavardė	Asmens pavardė. Privaloma nurodyti.	Atributas
El. pašto adresas	El. pašto adresas, kuriuo bus galima susisiekti. Neprivalomas nurodyti, jei yra nurodytas telefono numeris.	Atributas
Telefono numeris	Telefono numeris, kuriuo bus galima susisiekti. Neprivalomas nurodyti, jei nurodytas el. pašto adresas.	Atributas
Pareigos	Įmonės, kurioje dirba užsakovas pareigų pavadinimas arba nurodytos įmonės darbuotojo pareigų pavadinimas. Nurodyti nebūtinai.	Atributas

Duomenų modelio „Užsakovas“ klasės ir jos atributų aprašymas

Duomenų žodyno elemento pavadinimas	Turinys	Tipas
Užsakovas	Fizinis arba juridinis asmuo, užsakantis paslaugas.	Klasė
Atstovaujama įmonė	Įmonės pavadinimas kuriai atstovauja projekto užsakovas. Nurodyti nebūtina, jei užsakovas yra fizinis asmuo ir jokiai įmonei neatstovauja.	Atributas
Įmonės kodas	Įmonės kodas, kuriai atstovauja projekto užsakovas. Nurodyti nebūtina, jei užsakovas yra fizinis asmuo ir jokiai įmonei neatstovauja. Jeigu užsakovas dirba pagal individualią veiklą, tuomet rašomas individualios veiklos kodas.	Atributas

Duomenų modelio „Darbuotojas“ klasės ir jos atributų aprašymas

Duomenų žodyno elemento pavadinimas	Turinys	Tipas
Darbuotojas	Įmonėje dirbantis asmuo. Pavyzdžiui, projektų vadovas, dizaineris ar projektuotojas.	Klasė
Etatas	Kokiu etatu dirba darbuotojas. Nurodomas skaičius nuo 0 iki 1, kai 0 reiškia šiuo metu nedirba, 0.5 reiškia pusę etato, 1 – pilnas etatas, galimos ir kitos reikšmės jei patenka tarp 0 ir 1. Nurodyti būtina.	Atributas
Darbo patirtis	Kiek metų patirties darbuotojas turi dabartinėse ar susijusiose pareigose.	Atributas
Įdarbinimo data	Kada darbuotojas buvo įdarbintas įmonėje. Saugoma formatu YYYY-MM-DD. Nurodyti būtina.	Atributas

Duomenų modelio „Užduotis“ klasės ir jos atributų aprašymas

Duomenų žodyno elemento pavadinimas	Turinys	Tipas
Užduotis	Projektas yra išskaidomas į užduotis, kurias įgyvendina įmonės darbuotojai. Užduotys taip pat gali būti ir rastos klaidos, kurias reikia ištaisyti.	Klasė
Sukūrimo data ir laikas	Data ir laikas kada užduotis buvo įregistruota užduotis. Saugoma formatu YYYY-MM-DD HH:MM:SS. Reikšmė būtina.	Atributas
Atnaujinimo data ir laikas	Data ir laikas kada užduotis buvo atnaujinta. Saugoma formatu YYYY-MM-DD HH:MM:SS. Reikšmė būtina. Jeigu užduotis dar nebuvo niekad atnaujinta, tuomet atnaujinimo data ir laikas sutampa su sukūrimo data ir laiku.	Atributas
Aprašymas	Užduoties aprašymas arba užduoties sąlyga, paaiškinimas kas turi būti atlikta ar ištaisyta. Nurodyti būtina. Saugomas tekstas.	Atributas
Pagrindimas	Paaiškinimas kodėl užduotis turi būti atliekama. Nurodyti nebūtina.	Atributas
Nuotrauka	Jeigu užduoties paaiškinimui trūksta vaizdinės informacijos, tai ji įkeliama nuotraukos pavidalu. Įkelti nuotraukos nebūtina. Leidžiami formatai: jpg, png.	Atributas
Prioritetas	Nurodo kiek svarbi yra užduoties, pagal tai galima spręsti ar ją būtina įgyvendinti kuo greičiau ar galima dirbti normaliu režimu. Turi būti būtinai nurodytas vienas iš galimų prioritetų: žemas; normalus; aukštas; labai aukštas.	Atributas
Įvykdymo būseną	Nurodo kokia yra užduoties įvykdymo būseną. Turi būti būtinai nurodytas viena iš galimų būsenų: normali; vykdoma; baigta; atšaukta; vėluojanti.	Atributas
Terminas	Nurodo iki kada užduotis turi būti būtinai įgyvendinta. Nurodyti termino nebūtina, tokiu atveju bus laikoma, kad užduotis neturi termino ir ją galima atlikti kada gausis.	Atributas

2 priedas. Panaudos atvejų specifikacijos

Panaudos atvejo „Prisijungti“ specifikacija

Panaudojimo atvejis	PA-01. Prisijungti
Aprašymas	Neprisijungęs, bet registruotas vartotojas gali prisijungti prie sistemos.
Prieš-sąlyga	Vartotojas turi būti registruotas sistemoje.
Aktorius	Neprisijungęs vartotojas
Pagrindinis scenarijus	<pre> sequenceDiagram participant User as Neprisijungęs vartotojas participant System as Sistema User->>System: Suvesti duomenis activate System System->>System: Ar duomenys suvesti tinkamai? System-->>User: Ne deactivate System User->>System: Mato klaidos pranešimą activate System System->>System: Ar toks vartotojas egzistuoja? System-->>User: Taip deactivate System User->>System: Mato prieinamą informaciją activate System System->>System: Klaidos pranešimas System-->>User: deactivate System User->>System: Mato klaidos pranešimą deactivate System </pre>
Po-sąlyga	Vartotojas prijungiamas prie sistemos, gali matyti jam prieinamą informaciją ar atlikti kitus jam leidžiamus veiksmus.

Panaudos atvejo „Peržiūrėti naudojimo instrukcijas“ specifikacija

Panaudojimo atvejis	PA-02. Peržiūrėti naudojimo instrukcijas
Aprašymas	Bet kuris prisijungęs prie sistemos vartotojas gali peržiūrėti bet kurio funkcionalumo integruotas instrukcijas.
Prieš-sąlyga	Sistemoje, konkrečiam norimam funkcionalumo paaiškinimui turi būti įvesta instrukcijų (paaiškinimų).
Aktorius	Projekto užsakovas; sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram actor Vartotojas Vartotojas->>U1: Paspaudžia ant klaustuko ikonos activate U1 U1->>U2: Mato paaiškinimo tekstą activate U2 U2->>Vartotojas: deactivate U2 deactivate U1 </pre>
Po-sąlyga	Vartotojui yra parodomas pranešimas su paaiškinimais kaip naudotis konkrečiu funkcionalumu.

Panaudos atvejo „Atnaujinti projekto informaciją“ specifikacija

Panaudojimo atvejis	PA-03. Atnaujinti projekto informaciją
Aprašymas	Projekto užsakovas gali atnaujinti jam priklausančių projektų informaciją, įmonės darbuotojai (sistemos projektuotojai ir projektų vadovai) gali atnaujinti jiems priskirtų projektų informaciją, o sistemos administratorius gali atnaujinti bet kurio projekto informaciją.
Prieš-sąlyga	Projekto užsakovas turi turėti bent vieną jam priklausančią projektą, įmonės darbuotojas turi turėti bent vieną jam priskirtą projektą, sistemoje turi būti įvestas bent vienas projektas, kurį galėtų atnaujinti administratorius.
Aktorius	Projekto užsakovas; sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram actor Vartotojas participant Sistema Vartotojas->>Sistema: Projekto peržiūra activate Sistema Vartotojas->>Sistema: Paspaudimas ant projekto atnaujinimo deactivate Sistema Vartotojas->>Sistema: Duomenų įvedimas activate Sistema Sistema->>Sistema: Ar duomenys validūs? activate Sistema Sistema-->>Sistema: { } Sistema-->>Vartotojas: Mato klaidos pranešimą deactivate Sistema Vartotojas->>Sistema: Peržiūrėti naują informaciją activate Sistema Sistema->>Sistema: Išsaugoti pakeitimus deactivate Sistema Sistema-->>Vartotojas: Peržiūrėti naują informaciją deactivate Sistema Vartotojas->>Vartotojas: End </pre>
Po-sąlyga	Projekto informacija yra atnaujinama, pakeitimus gali matyti kiti naudotojai, kurie prieina prie to projekto.

Panaudos atvejo „Peržiūrėti projekto prototipų sąrašą“ specifikacija

Panaudojimo atvejis	PA-04. Peržiūrėti projekto prototipų sąrašą
Aprašymas	Viename projekte gali būti sukurti keli prototipai, kurie gali būti skirtingo sudėtingumo lygio arba realizuojantys skirtingas idėjas. Projekto užsakovas arba prie projekto priskirti kiti vartotojai gali peržiūrėti tokių projekte esančių prototipų sąrašą.
Prieš-sąlyga	Turi būti sukurtas bent vienas projektas, kuriame būtų galima žiūrėti prototipų sąrašą. Vartotojui turi priklausyti arba būti priskirtas projektas, kurio prototipus yra bandoma peržiūrėti.
Aktorius	Projekto užsakovas; sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram participant V as Vartotojas participant S as Sistema V->>S: Eiti į projektą S->>S: Ar projektas turi prototipų? S-->>S: Ne S-->>S: Taip S->>V: Atrinkti prototipus atvaizdavimui V->>S: Mato prienamą informaciją </pre> <p>The diagram illustrates the interaction between a user (Vartotojas) and a system (Sistema) to view a project's prototype list. The process starts with the user initiating the action 'Eiti į projektą' (Go to project). The system then checks 'Ar projektas turi prototipų?' (Does the project have prototypes?). A decision diamond follows, with 'Ne' (No) leading back to the user and 'Taip' (Yes) leading to 'Atrinkti prototipus atvaizdavimui' (Select prototypes for display). The system then sends 'Atrinkti prototipus atvaizdavimui' to the user, who receives 'Mato prienamą informaciją' (Views available information) and ends the process.</p>
Po-sąlyga	Matomas projekto prototipų sąrašas arba pranešimas, kad prototipų projekte dar nėra sukurta.

Panaudos atvejo „Peržiūrėti projektų sąrašą“ specifikacija

Panaudojimo atvejis	PA-05. Peržiūrėti projektų sąrašą
Aprašymas	Projekto užsakovas gali peržiūrėti jam priklausančių projektų sąrašą. Sistemos projektuotojas, projektų vadovas arba sistemos administratorius gali peržiūrėti bet kurį sistemoje esantį projektą.
Prieš-sąlyga	Sistemoje turi būti sukurtas bent vienas projektas.
Aktorius	Projekto užsakovas; sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	
Po-sąlyga	Atvaizduojamas projektų sąrašas arba žinutė, kad projektų nerasta.

Panaudos atvejo „Peržiūrėti prototipo langų sąrašą“ specifikacija

Panaudojimo atvejis	PA-06. Peržiūrėti prototipo langų sąrašą
Aprašymas	Peržiūrint projekto prototipą galima matyti jame visų esančių langų sąrašą. Projekto užsakovas, kuriam priklauso projektas arba kitas sistemos naudotojas, kuriam priskirtas projektas, gali matyti tokį prototipo langų sąrašą.
Prieš-sąlyga	Turi būti sukurtas bent vienas projektas, o jame bent vienas prototipas. Vartotojui turi priklausyti arba būti priskirtas projektas, kuriame yra prototipas.
Aktorius	Projekto užsakovas; sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	
Po-sąlyga	Vartotojui yra rodomas projekto prototipo langų sąrašas.

Panaudos atvejo „Peržiūrėti pasirinktą prototipo langą“ specifikacija

Panaudojimo atvejis	PA-07. Peržiūrėti pasirinktą prototipo langą
Aprašymas	Projekto užsakovas, kuriam priklauso projektas arba sistemos vartotojas, kuriam priskirtas projektas, gali peržiūrėti bet kurį projekto prototipo langą.
Prieš-sąlyga	Turi būti sukurtas bent vienas projektas, kuriame būtų bent vienas prototipas, kuriame būtų bent vienas langas. Vartotojui turi priklausyti arba būti priskirtas projektas, kurio prototipo langą bandoma peržiūrėti.
Aktorius	Projekto užsakovas; sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	
Po-sąlyga	Atvaizduojamas projekto prototipo langas.

Panaudos atvejo „Peržiūrėti visus sugeneruotus prototipus“ specifikacija

Panaudojimo atvejis	PA-08. Peržiūrėti visus sugeneruotus prototipus
Aprašymas	Vartotojas su atitinkama role gali peržiūrėti visus sistemoje esančius prototipus ir jų langus.
Prieš-sąlyga	Vartotojas turi turėti projektuotojo, projektų vadovo arba sistemos administratoriaus rolę.
Aktorius	Sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	
Po-sąlyga	Atvaizduojamas visų sistemoje esančių prototipų sąrašas.

Panaudos atvejo „Peržiūrėti prototipo lango pasiūlymus“ specifikacija

Panaudojimo atvejis	PA-09. Peržiūrėti prototipo lango pasiūlymus
Aprašymas	Vartotojas su tinkama role, peržiūrėdamas prototipo langą gali matyti papildomų pasiūlymų kaip kitaip tas langas galėtų būti atvaizduojamas.
Prieš-sąlyga	Turi būti sukurtas bent vienas projektas, kuriame turėtų būti bent vienas prototipas, kuriame bent vienas langas. Vartotojas turi turėti projektuotojo, projektų vadovo arba sistemos administratoriaus rolę.
Aktorius	Sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram participant V as Vartotojas participant S as Sistema V->>V: Peržiūri prototipo langą V->>S: Paspaudžia gauti pasiūlymų S->>S: Sugeneruoja kelis pasiūlymus S->>V: Parengia sugeneruotus langus peržiūrai V->>V: Mato rezultatus </pre>
Po-sąlyga	Vartotojui atvaizduojami papildomi pasiūlymai konkretaus lango vartotojo sąsajos išdėstymui.

Panaudos atvejo „Sukurti prototipą“ specifikacija

Panaudojimo atvejis	PA-10. Sukurti prototipą
Aprašymas	Vartotojas su atitinkama role gali sukurti naują prototipą projekte.
Prieš-sąlyga	Sistemoje turi būti sukurtas bent vienas projektas. Vartotojas turi turėti projektuotojo, projektų vadovo arba sistemos administratoriaus rolę.
Aktorius	Sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram participant V as Vartotojas participant S as Sistema V->>S: Suvesti duomenis S->>S: Ar duomeys validūs? S-->>V: Ne S->>V: Taip S->>S: Sukurti prototipą su pasirinkta informacija S->>V: Peržiūri naujai sukurtą prototipą </pre>
Po-sąlyga	Atvaizduojamas sukurtas prototipas.

Panaudos atvejo „Atnaujinti prototipo informaciją“ specifikacija

Panaudojimo atvejis	PA-11. Atnaujinti prototipo informaciją
Aprašymas	Vartotojas su atitinkama role gali atnaujinti prototipo informaciją, tokią kaip pavadinimas, aprašymas ar būseną.
Prieš-sąlyga	Turi būti sukurtas bent vienas prototipas projekte. Vartotojas turi turėti projektuotojo, projektų vadovo arba sistemos administratoriaus rolę.
Aktorius	Sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram actor User participant System User->>System: Prototipo peržiūra activate System System->>User: Užklausa dėl atnaujinimo deactivate System User->>System: Duomenų suvedimas activate System System->>System: Ar duomenys validūs? System-->>System: {Ar duomenys validūs?} System-->>User: Mato klaidos pranešimą deactivate System User->>System: Prototipo peržiūra activate System System->>System: Išsaugoti informaciją deactivate System System-->>User: Prototipo peržiūra deactivate System </pre>
Po-sąlyga	Prototipo informacija yra atnaujinama, atvaizduojamas atnaujintas prototipas.

Panaudos atvejo „Šalinti prototipą“ specifikacija

Panaudojimo atvejis	PA-14. Šalinti prototipą
Aprašymas	Vartotojas su atitinkama role gali pašalinti nebereikalingą prototipą iš projekto.
Prieš-sąlyga	Projekte turi būti sukurtas bent vienas prototipas. Vartotojas turi turėti projektuotojo, projektų vadovo arba sistemos administratoriaus rolę.
Aktorius	Sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	
Po-sąlyga	Prototipas pašalinamas. Atvaizduojamas projekte esančių prototipų sąrašas.

Panaudos atvejo „Šalinti prototipo langą“ specifikacija

Panaudojimo atvejis	PA-15. Šalinti prototipo langą
Aprašymas	Vartotojas gali pašalinti langą iš prototipo.
Prieš-sąlyga	Prototipe turi būti bent vienas langas, kurį eiti šalinti.
Aktorius	Sistemos projektuotojas; projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	
Po-sąlyga	Prototipo langas pašalinamas iš sistemos. Atvaizduojamas prototipo langų sąrašas.

Panaudos atvejo „Sukurti projektą“ specifikacija

Panaudojimo atvejis	PA-16. Sukurti projektą
Aprašymas	Projektų vadovas arba administratorius gali sukurti projektą su kuriuo bus galima susieti užsakovą bei prototipus.
Prieš-sąlyga	Vartotojas turi turėti projektų vadovo arba administratoriaus rolę.
Aktorius	Projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram actor Vartotojas participant Sistema Vartotojas->>Sistema: Suvesti duomenis activate Sistema Sistema->>Sistema: Ar duomenys validūs? activate Sistema Sistema-->>Vartotojas: Ne deactivate Sistema Vartotojas->>Sistema: Suvesti duomenis activate Sistema Sistema->>Sistema: Sukuriamas projektas activate Sistema Sistema-->>Vartotojas: Projekto informacijos peržiūra deactivate Sistema Vartotojas->>Vartotojas: Projekto informacijos peržiūra deactivate Vartotojas </pre> <p>The diagram illustrates the process of creating a project. It starts with the user (Vartotojas) providing data (Suvesti duomenis). The system (Sistema) then checks if the data is valid (Ar duomenys validūs?). If not (Ne), the user receives an error message (Klaidos pranešimas) and must re-enter the data. If yes (Taip), the system creates the project (Sukuriamas projektas), and the user can then view the project information (Projekto informacijos peržiūra).</p>
Po-sąlyga	Atvaizduojamas projektas ir jo informacija.

Panaudos atvejo „Priskirti projektą prie užsakovo“ specifikacija

Panaudojimo atvejis	PA-17. Priskirti projektą prie užsakovo
Aprašymas	Projektų vadovas arba administratorius gali priskirti projektą prie užsakovo.
Prieš-sąlyga	Turi būti užsakovo vartotojo paskyra ir sukurtas projektas.
Aktorius	Projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	
Alternatyvus scenarijus	
Po-sąlyga	Projektas susiejamas su užsakovu. Užsakovas gali peržiūrėti projektą. Atvaizduojamas projektas.

Panaudos atvejo „Archyvuoti projektą“ specifikacija

Panaudojimo atvejis	PA-18. Archyvuoti projektą
Aprašymas	Projektų vadovas arba administratorius gali archyvuoti projektą, kuris seniai pasibaigęs.
Prieš-sąlyga	Turi būti sukurtas bent vienas projektas, kurį būtų galima archyvuoti.
Aktorius	Projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	
Po-sąlyga	Projektas suarchyvuojamas, paslepiaama jo peržiūra iš bendro projektų sąrašo. Atvaizduojamas bendras projektų sąrašas.

Panaudos atvejo „Peržiūrėti suarchyvuotus projektus“ specifikacija

Panaudojimo atvejis	PA-19. Peržiūrėti suarchyvuotus projektus
Aprašymas	Projektų vadovas arba administratorius gali peržiūrėti archyvuotų projektų sąrašą.
Prieš-sąlyga	Vartotojas turi turėti projektų vadovo arba sistemos administratoriaus rolę.
Aktorius	Projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	

Po-sąlyga	Atvaizduojamas archyvuotų projektų sąrašas.
------------------	---

Panaudos atvejo „Atstatyti projektą iš archyvo“ specifikacija

Panaudojimo atvejis	PA-20. Atstatyti projektą iš archyvo
Aprašymas	Projektų vadovas arba administratorius gali prikelti projektą iš archyvo.
Prieš-sąlyga	Vartotojas turi turėti projektų vadovo arba sistemos administratoriaus rolę.
Aktorius	Projektų vadovas; sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram actor Vartotojas participant Sistema Vartotojas->>Sistema: Paklausti ar tikrai atstatyti activate Sistema Sistema->>Vartotojas: Vartotojas pasirinko atkūrimą deactivate Sistema Vartotojas->>Sistema: Vartotojas atšaukė veiksmą activate Sistema Sistema->>Vartotojas: Projektas atstatomas iš archyvo deactivate Sistema Vartotojas->>Sistema: Pasirinkti atstatymą iš archyvo activate Sistema Sistema->>Vartotojas: Projektas atstatomas iš archyvo deactivate Sistema Vartotojas->>Sistema: Pasirinkti projektą activate Sistema Sistema->>Vartotojas: Projektas atstatomas iš archyvo deactivate Sistema Vartotojas->>Sistema: Archyvuotų projektų peržiūra activate Sistema Sistema->>Vartotojas: Projektas atstatomas iš archyvo deactivate Sistema </pre> <p>The diagram illustrates the process of restoring a project from an archive. It is divided into two swimlanes: 'Vartotojas' (User) and 'Sistema' (System). The process starts with the user viewing archived projects, selecting one, and then choosing to restore it from the archive. The system then asks for confirmation. A decision diamond follows, where the user can either confirm the restoration (leading to the project being restored) or abort the action (leading back to the selection step). The process ends with the user viewing the restored project.</p>
Po-sąlyga	Projektas pradamas vėl rodyti bendrame projektų sąrašė. Atvaizduojamas projektas.

Panaudos atvejo „Atnaujinti vartotojo rolę“ specifikacija

Panaudojimo atvejis	PA-21. Atnaujinti vartotojo rolę
Aprašymas	Sistemos administratorius gali pakeisti bet kurio kito vartotojo rolę.
Prieš-sąlyga	Vartotojas turi turėti sistemos administratoriaus rolę. Sistemoje turi būti bent vienas kitas registruotas vartotojas.
Aktorius	Sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram actor Actor as Vartotojas actor System as Sistema Actor->>Actor: Vartotojo peržiūra Actor->>Actor: Vartotojo atnaujinimo pasirinkimas Actor->>Actor: Vartotojo rolės keitimo pasirinkimas Actor->>Actor: Rolės pasirinkimas Actor->>System: Ar tikrai galima keisti rolę? System-->>Actor: Klaidos pranešimo peržiūra (Ne) System-->>Actor: Atnaujinti rolę (Taip) Actor->>Actor: Vartotojo peržiūra </pre>
Po-sąlyga	Vartotojo rolę pakeičiama. Atvaizduojama vartotojo informacija.

Panaudos atvejo „Registruoti naują vartotoją“ specifikacija

Panaudojimo atvejis	PA-22. Registruoti naują vartotoją
Aprašymas	Sistemos administratorius gali registruoti naują vartotoją sistemoje.
Prieš-sąlyga	Vartotojas turi turėti sistemos administratoriaus rolę.
Aktorius	Sistemos administratorius
Pagrindinis scenarijus	<pre> sequenceDiagram participant V as Vartotojas participant S as Sistema V->>S: Suvesti duomenis S->>S: Ar duomenys validūs? S-->>V: Ne S-->>V: Taip S->>V: Sukurto vartotojo peržiūra V->>S: Klaidos pranešimo peržiūra S->>S: Sukuriamas naujas vartotojas </pre> <p>The diagram illustrates the process of registering a new user. It is divided into two swimlanes: 'Vartotojas' (User) and 'Sistema' (System). The process starts with the user providing data ('Suvesti duomenis'). The system then checks if the data is valid ('Ar duomenys validūs?'). If not valid ('Ne'), the user is notified of the error ('Klaidos pranešimo peržiūra'). If valid ('Taip'), a new user is created ('Sukuriamas naujas vartotojas'), and the user is shown the newly created user ('Sukurto vartotojo peržiūra'). The process ends with the user reviewing the error message ('Klaidos pranešimo peržiūra').</p>
Po-sąlyga	Vartotojas užregistruojamas sistemoje. Atvaizduojama vartotojo informacija.

3 priedas. Nefunkciniai reikalavimai

REIKALAVIMAI SISTEMOS IŠVAIZDAI

Išvaizdos reikalavimai

Reikalavimas #:	NF-01	Reikalavimo tipas:	V10	PA/FR #:	N/A
Aprašymas:	Sistemos grafinėje vartotojo sąsajoje visi grafiniai elementai ant kurių galima spausti turėtų būti sąlyginai dideli.				
Pagrindimas:	Vartotojai naudosis sistema nebūtina sėdint priešais kompiuterio ekraną, todėl jiems turi būti patogų matyti grafinius elementus, tokius kaip mygtukai ir iš toliau.				
Šaltinis:	Įmonės MB „Projektavimas“ direktorius Edvinas Jankauskis				
Atitikimo kriterijus:	Reikalavimas bus įgyvendintas jeigu bent 70 proc. testuotų sistemos naudotojų patvirtins, kad sėdint nuo 21“ įstrižainės monitoriaus per 50-100 cm, tai nesukėlė nepatogumų naudotis sistema.				
Užsakovo pasitenkinimas:	4	Užsakovo nepasitenkinimas:	3		
Prioritetas:	Žemas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-05				

Stiliaus reikalavimai

Reikalavimas #:	NF-03	Reikalavimo tipas:	V10	PA/FR #:	N/A
Aprašymas:	Sistemos grafinė vartotojo sąsaja turi jaustis neapkrauta.				
Pagrindimas:	Pagrindinis vartotojų dėmesys turėtų būti skirtas pradinių duomenų įkėlimui bei apdorotų rezultatų peržiūrai, todėl pačios sistemos grafinė vartotojo sąsaja turėtų būti neapkrauta ir turėtų neblaškyti naudotojų dėmesio.				
Šaltinis:	Įmonės MB „Projektavimas“ technologijų vadovas Audrius Marcevičius				
Atitikimo kriterijus:	Sąsaja jausis lengva jei vidutiniškai kiekviename lange bus mažiau nei 10 galimų parinkčių ant ko galima spausti norint atlikti kokius nors veiksmus. Išimtis būtų įvedimo formos, skirtos administratoriui, nes jis tikriausiai sėdės priešais kompiuterį.				
Užsakovo pasitenkinimas:	4	Užsakovo nepasitenkinimas:	3		
Prioritetas:	Aukštas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-05				

REIKALAVIMAI PANAUDOJAMUMUI

Suprantamumo ir mandagumo reikalavimai

Reikalavimas #:	NF-06	Reikalavimo tipas:	V11	PA/FR #:	N/A
Aprašymas:	Sistemoje naudojama terminologija turėtų būti aiški tiek projektų užsakovams, tiek įmonėms darbuotojams.				

Pagrindimas:	Naudojama nesuprantama terminologija gali pabloginti sistemos naudojamumo patirtį ir vartotojams gali būti sunkiau rasti kaip atlikti norimas užduotis.		
Šaltinis:	Įmonės MB „Projektavimas“ marketingo vadovas Mindaugas Viskalas		
Atitikimo kriterijus:	Iš visų testuotų vartotojų bent 70 proc. jų turėtų teigti, kad terminologija jiems nesukėlė jokių problemų suprasti kur ir ką galima atlikti.		
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	3
Prioritetas:	Žemas	Konfliktai:	Nėra
Istorija:	Sukurta 2020-03-09		

Suteikiami patogumai

Reikalavimas #:	NF-07	Reikalavimo tipas:	V11	PA/FR #:	N/A
Aprašymas:	Sistema turėtų informuoti apie naujus prototipus projekte, naujus langus prototipe ir kitus atnaujinimus, kurie gali būti aktualūs konkrečiam vartotojo tipui.				
Pagrindimas:	Vartotojai dažnai tikisi, kad automatizuota sistema turės ir suteikiamų patogumų, tokių kaip pranešimai apie naujus įvykius, kurie yra susiję su tuo vartotoju. Norint pakelti vartotojų pasitenkinimą sistema, reikėtų turėti vartotojų informavimo galimybes.				
Šaltinis:	Įmonės MB „Projektavimas“ marketingo vadovas Mindaugas Viskalas				
Atitikimo kriterijus:	Vartotojai gaus pranešimus apie pasikeitimus jų projektuose.				
Užsakovo pasitenkinimas:	3	Užsakovo nepasitenkinimas:	1		
Prioritetas:	Žemas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-09				

EFEKTYVUMO IR NAŠUMO REIKALAVIMAI

Reikalavimai užduočių vykdymo greičiui

Reikalavimas #:	NF-08	Reikalavimo tipas:	V12	PA/FR #:	N/A
Aprašymas:	Bet kuri užduotis sistemoje turi duoti atsaką greičiau nei per 2 sekundes.				
Pagrindimas:	Sistema turi veikti greitai, kad netrukdytų įmonės darbuotojų darbo. Jei pasitaikys tokių užduočių, kurios gali užtrukti ilgiau, pavyzdžiui, prototipo generavimas, tuomet per tas 2 sekundes ar greičiau turi būti pradėta rodyti progreso juosta.				
Šaltinis:	Įmonės MB „Projektavimas“ direktorius Edvinas Jankauskis				
Atitikimo kriterijus:	Sistema į bet kokią užklausą atsako greičiau nei per 2 sekundes, o ilgiau užtrunkančių užduočių atveju per tą patį laiką pradeda rodyti progreso juostą.				
Užsakovo pasitenkinimas:	4	Užsakovo nepasitenkinimas:	3		
Prioritetas:	Vidutinis	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-09				

Reikalavimai tikslumui

Reikalavimas #:	NF-09	Reikalavimo tipas:	V12	PA/FR #:	PA-12
Aprašymas:	Gauti prototipų generavimo rezultatai turėtų būti gana tikslūs.				
Pagrindimas:	Netiksliai veikianti sistema atbaidys įmonės darbuotojus nuo naudojimo ja.				
Šaltinis:	Ieva Jakšaitytė				
Atitikimo kriterijus:	Gauti prototipų generavimo rezultatai turėtų visada atitikti bent 90 proc. tai ką omenyje turėjo sistemos projektuodamas, piešdamas lango eskizą. Testuojama turėtų būti bent 20 valandų, kad būtų galima nuspręsti ar atitinka reikalavimus.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	5		
Prioritetas:	Aukštas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-09				

Reikalavimai produkto ilgaamžiškumui

Reikalavimas #:	NF-11	Reikalavimo tipas:	V12	PA/FR #:	N/A
Aprašymas:	Sistema, neviršijant biudžeto turėtų dirbti bent 2 metus nuo jos paleidimo.				
Pagrindimas:	Įmonė norėtų neviršyti numatomo biudžeto.				
Šaltinis:	Įmonės MB „Projektavimas“ direktorius Edvinas Jankauskis				
Atitikimo kriterijus:	Kriterijus bus įgyvendintas jei kiekvieną mėnesį dviejų metų laikotarpyje nuo sistemos paleidimo ant sistemos išlaikymo bus išleidžiama ne daugiau nei numatyta apribojimuose.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	3		
Prioritetas:	Vidutinis	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-09				

REIKALAVIMAI VEIKIMO SĄLYGOMS

Reikalavimai leidybos procesui

Reikalavimas #:	NF-12	Reikalavimo tipas:	V13	PA/FR #:	N/A
Aprašymas:	Kiekviena nauja sistemos versija turėtų nesugadinti duomenų, esančių sistemoje.				
Pagrindimas:	Sistemoje saugomi duomenys visai svarbus ją naudojantiems vartotojams, todėl tai neturėtų dingti ar susigadinti atnaujinimo metu.				
Šaltinis:	Įmonės MB „Projektavimas“ technologijų vadovas Audrius Marcevičius				
Atitikimo kriterijus:	Reikalavimas bus įgyvendintas jei atnaujinimo metu duomenys nedings ir nesusigadins.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	4		

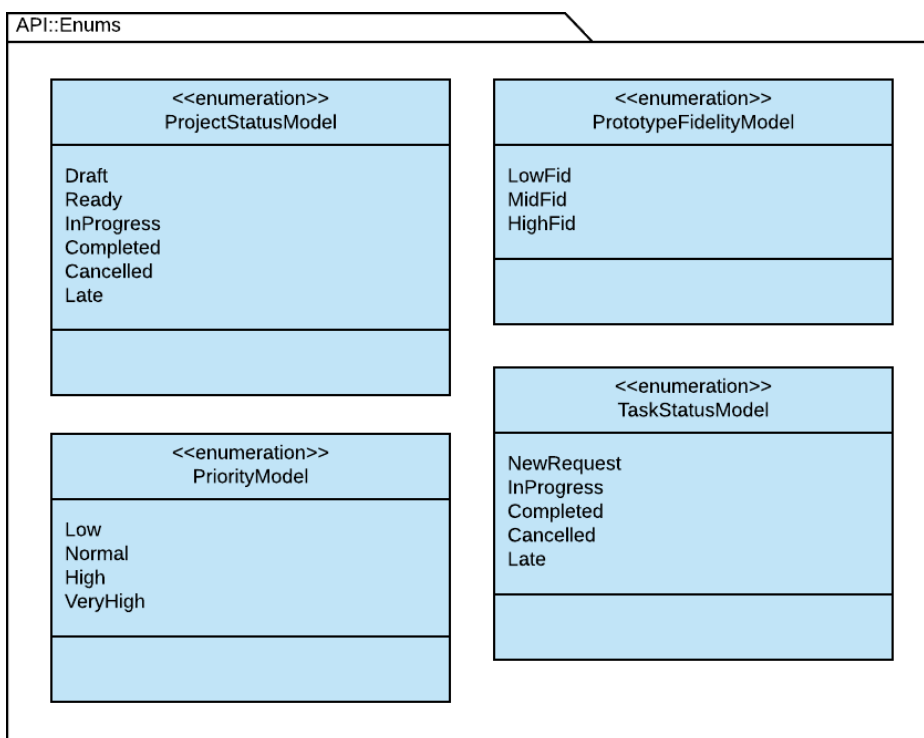
Prioritetas:	Aukštas	Konfliktai:	Nėra
Istorija:	Sukurta 2020-03-06		

SAUGUMO REIKALAVIMAI

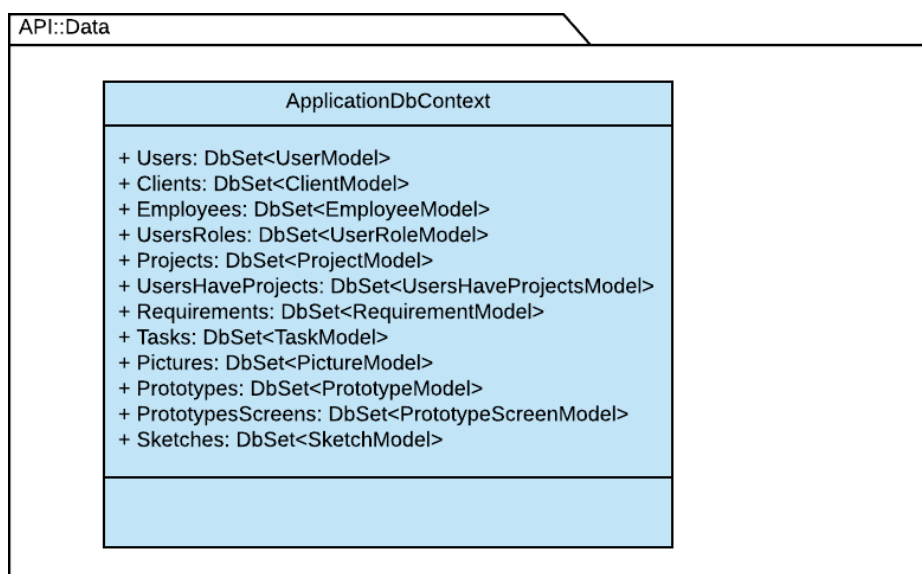
Reikalavimas #:	NF-14	Reikalavimo tipas:	V15	PA/FR #:	N/A
Aprašymas:	Tik įgaliotieji asmenys gali peržiūrėti sistemoje registruotus užsakovus ar kitus vartotojus.				
Pagrindimas:	Kai kurie klientai ar jų projektai gali būti laikomi konfidencialiais, todėl norint supaprastinti jų informacijos apsaugojimą, paprasti vartotojai neturėtų galėti pasiekti kitų vartotojų informacijos išvis.				
Šaltinis:	Įmonės MB „Projektavimas“ direktorius Petras Jankauskis				
Atitikimo kriterijus:	Yra sistemos administratoriaus rolė. Tik šią rolę turintis vartotojas gali matyti užsakovų sąrašą.				
Užsakovo pasitenkinimas:	5	Užsakovo nepasitenkinimas:	4		
Prioritetas:	Aukštas	Konfliktai:	Nėra		
Istorija:	Sukurta 2020-03-08				

4 priedas. Detalizuotos paketų diagramos

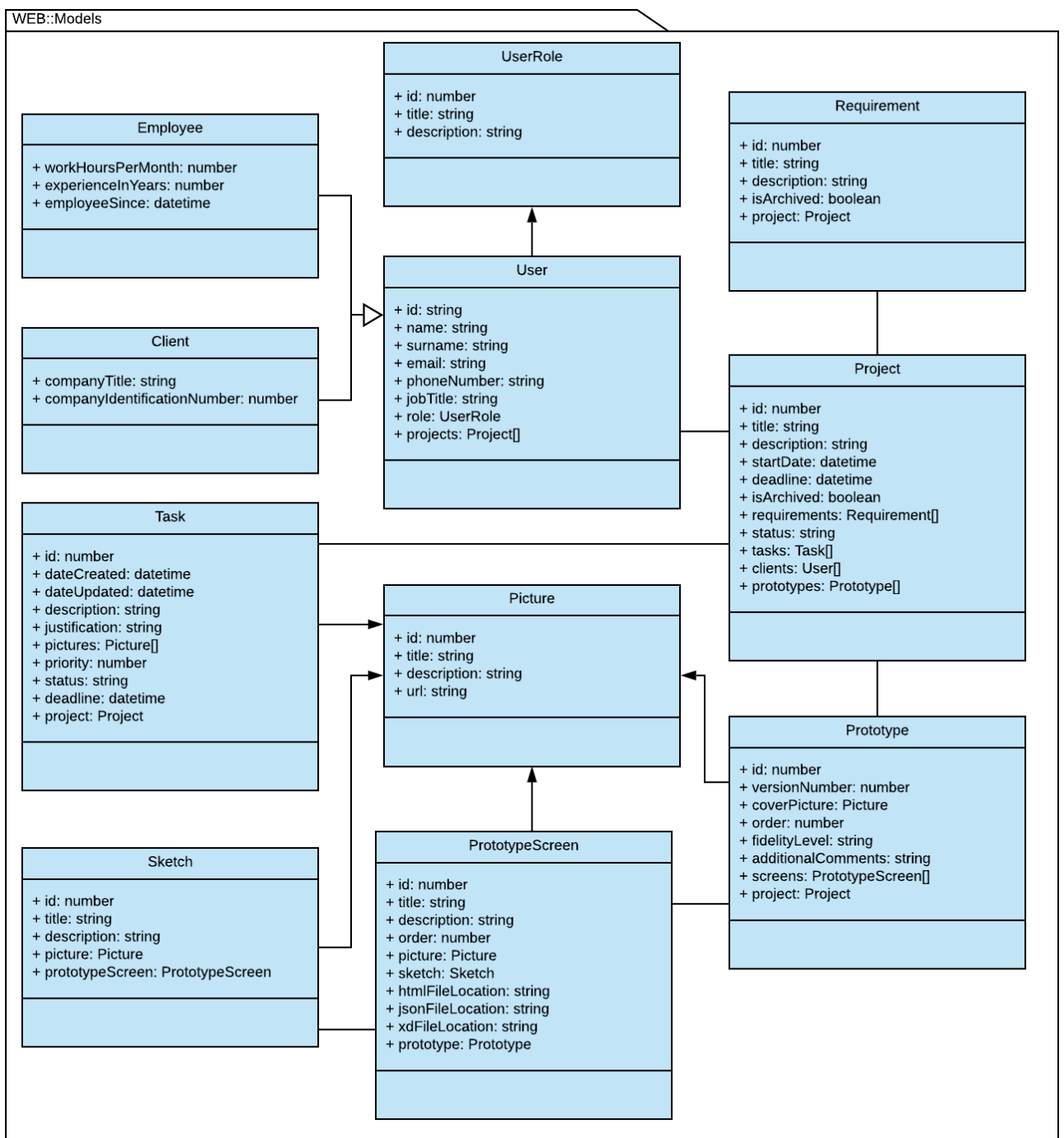
Paketo *API::Enums* klasių diagrama matoma žemiau. Šios klasės bus naudojamos *API::Models* ir *API::Controllers* paketų klasėse. Tai padės apibrėžti tam tikras duomenų reikšmes, tokias kaip tam tikra projekto būsena ar keli galimi prioritetų lygiai. Šie *Enums* bus naudojami kai kuriuose modelių klasėse norint priskirti tam tikras reikšmes prie kažkurių klasės atributų, taip pat, bus naudojami kontrolieriuose norint atfiltruoti reikiamus duomenis.



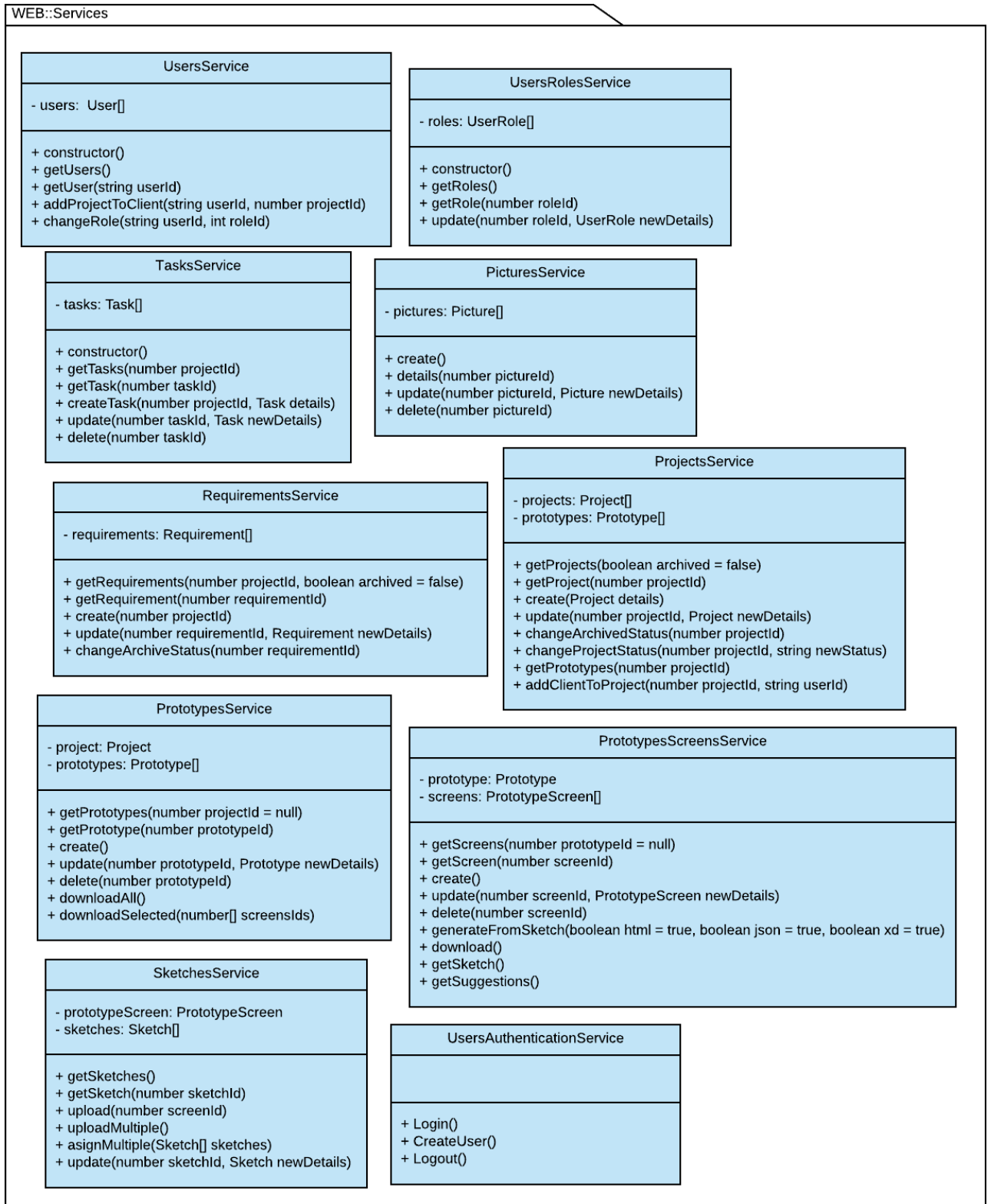
Paketo *API::Data* klasių diagrama matoma žemiau. Šio paketo klasė *ApplicationDbContext* bus naudojama norint susieti *API::Models* paketo klasių atributus su duomenų bazės lentelių stulpeliais. Taip pat, ši klasė bus naudojama *API::Controllers* klasėse kaip priklausomybių injekcija (angl. *Dependency Injection*) ir taip padės pasiekti duomenų bazę naudojant programoje aprašytų duomenų modelių klasių atributus.



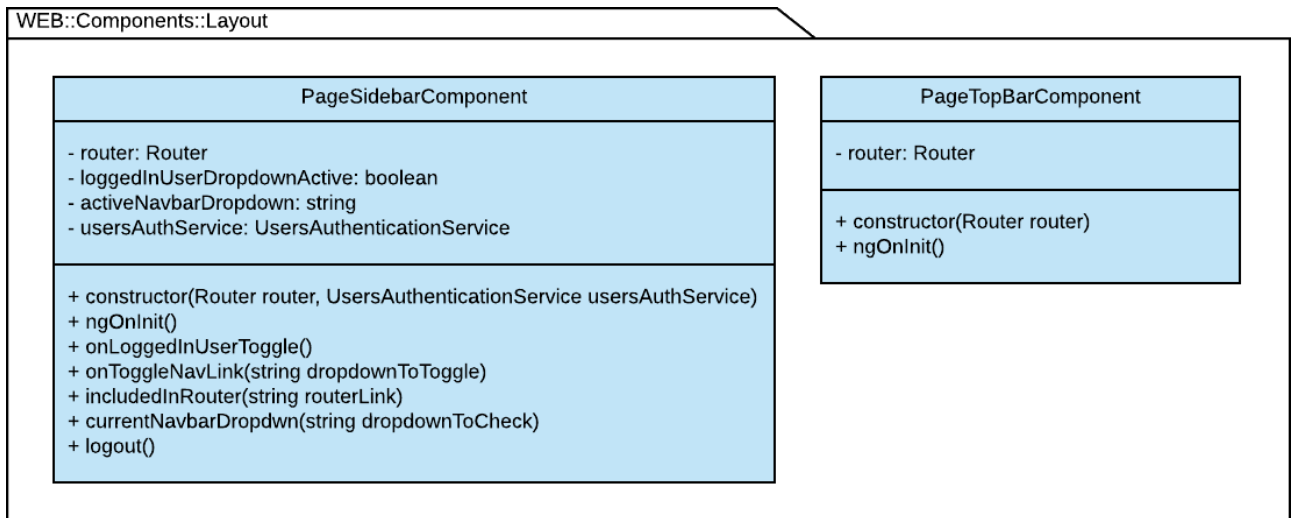
Paketo *WEB::Models* klasių diagrama matoma žemiau. Šios klasės bus naudojamos WEB aplikacijoje. Didžioji dalis klasių išlieka tokios pačios ar labai panašios kaip ir *API::Models* pakete, tačiau yra tokių skirtumų kaip skirtingi kintamųjų tipai. Modeliai, nurodyti šiame pakete bus naudojami visur kur jų reikės WEB aplikacijoje, pavyzdžiui, komponentuose ar servisuose. Taip pat, yra esminių skirtumų tarp *WEB::Models* ir *API::Models* dėl naudojamų skirtingų programavimo kalbų. API dalyje bus naudojamas C#, kuriame viskas yra aprašoma tvirtai (angl. *Strongly typed*), o WEB aplikacijai bus naudojamas karkasas paremtas *JavaScript* pagrindu, o pats *JavaScript* yra aprašomas laisvai (angl. *Loosely typed*). Todėl paketo *WEB::Models* klasių diagramos reikėtų nepriimti tiesiogiai, nes galutiniai objektai gali atrodyti ir kitaip nei nurodyta žemiau esančioje diagramoje, nes juos galima dinamiškai papildyti ar apkarpyti su reikiamais atributais programos eigoje.



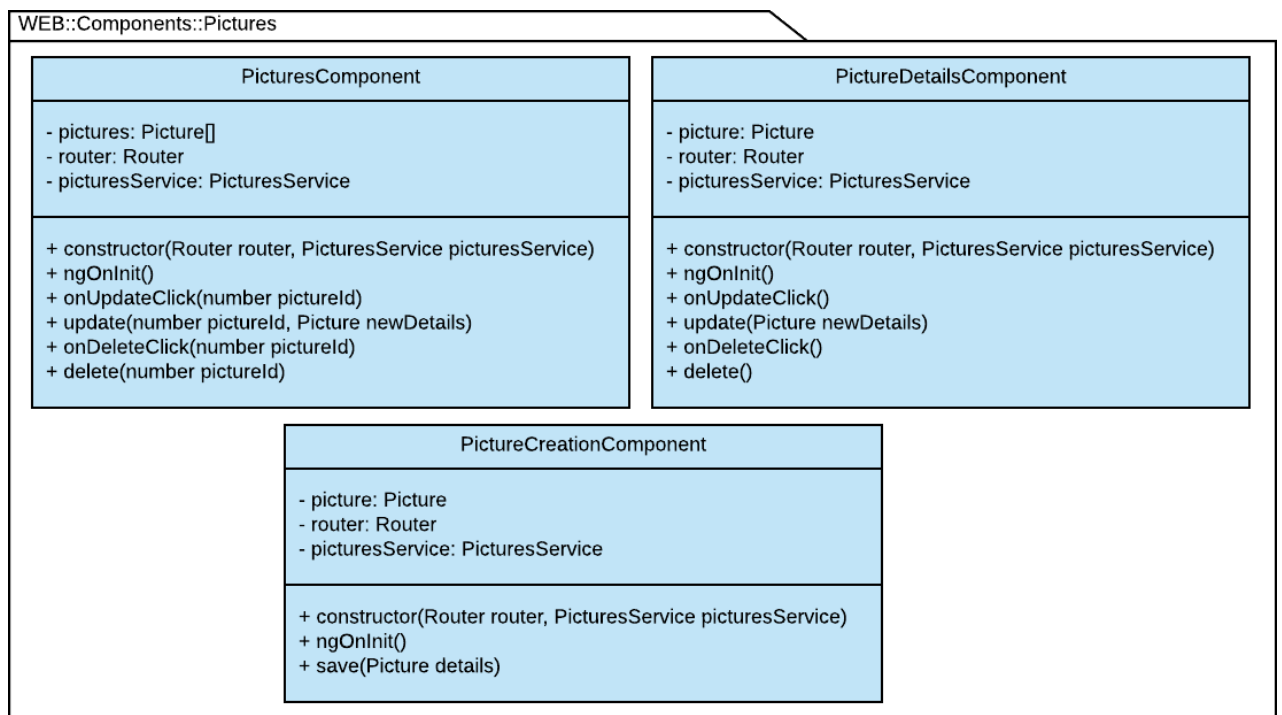
Paketo *WEB::Services* klasių diagrama matoma žemiau. Šio paketo klasių pagrindinė užduotis tai kauptis su atitinkamomis užklausomis į atitinkamas *API::Controllers* klases, ar atlikti bendrus veiksmus su gautais duomenimis iš *API*. Šios klasės per priklausomybių injekciją (angl. *Dependency Injection*) bus naudojamos *WEB::Components* paketo klasėse. Šio paketo klasės, ten kur reikia naudoja atitinkamus modelius iš *WEB::Models* paketo.



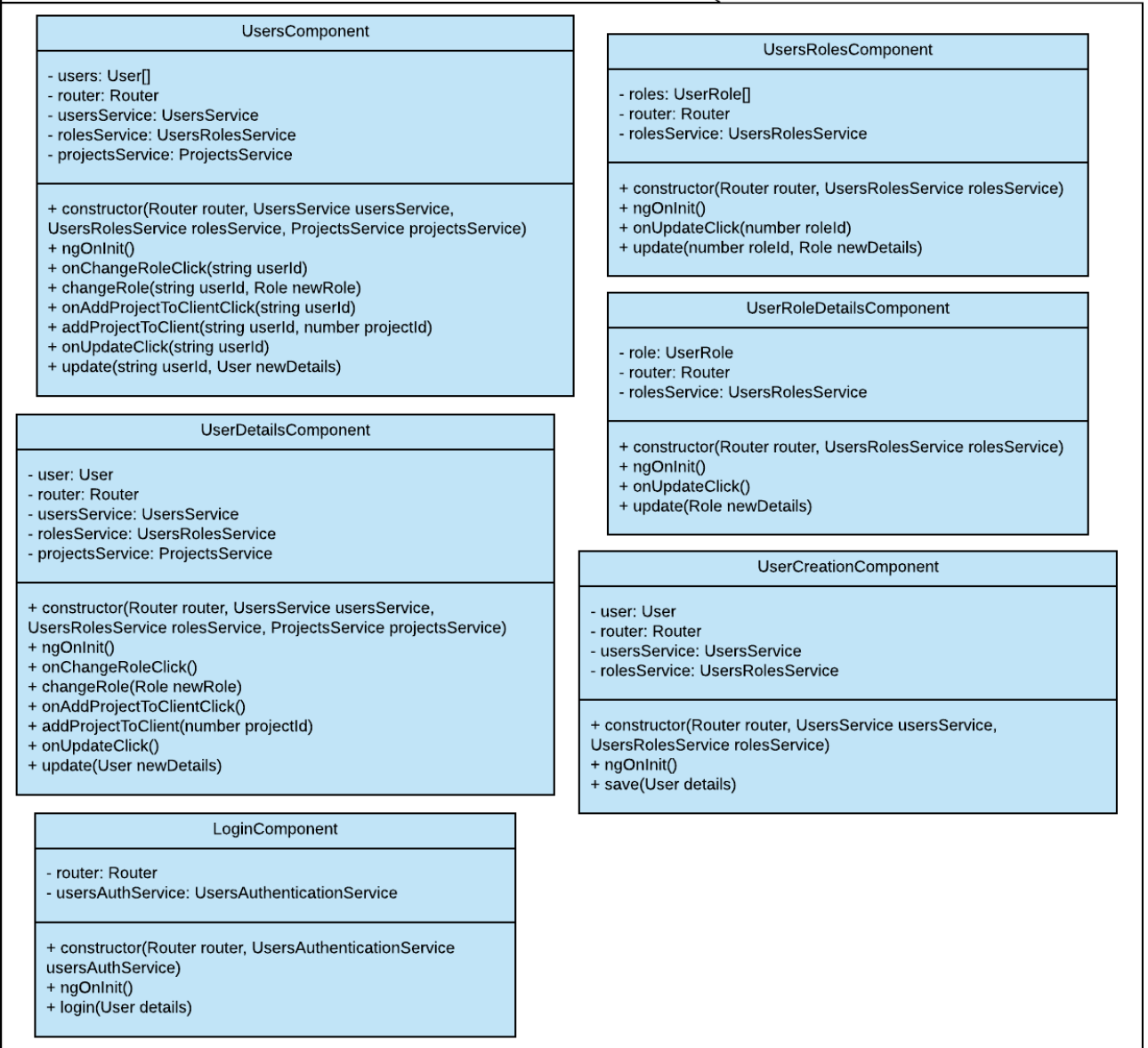
Paketo *WEB::Components::Layout* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos kitų *WEB::Components* paketų klasių. Šios klasės yra naudojamos norint sukurti atsikartojančius komponentus sistemos vartotojo sąsajoje, tokius kaip šoninė įrankių juosta ar viršutinė informacinė juosta.



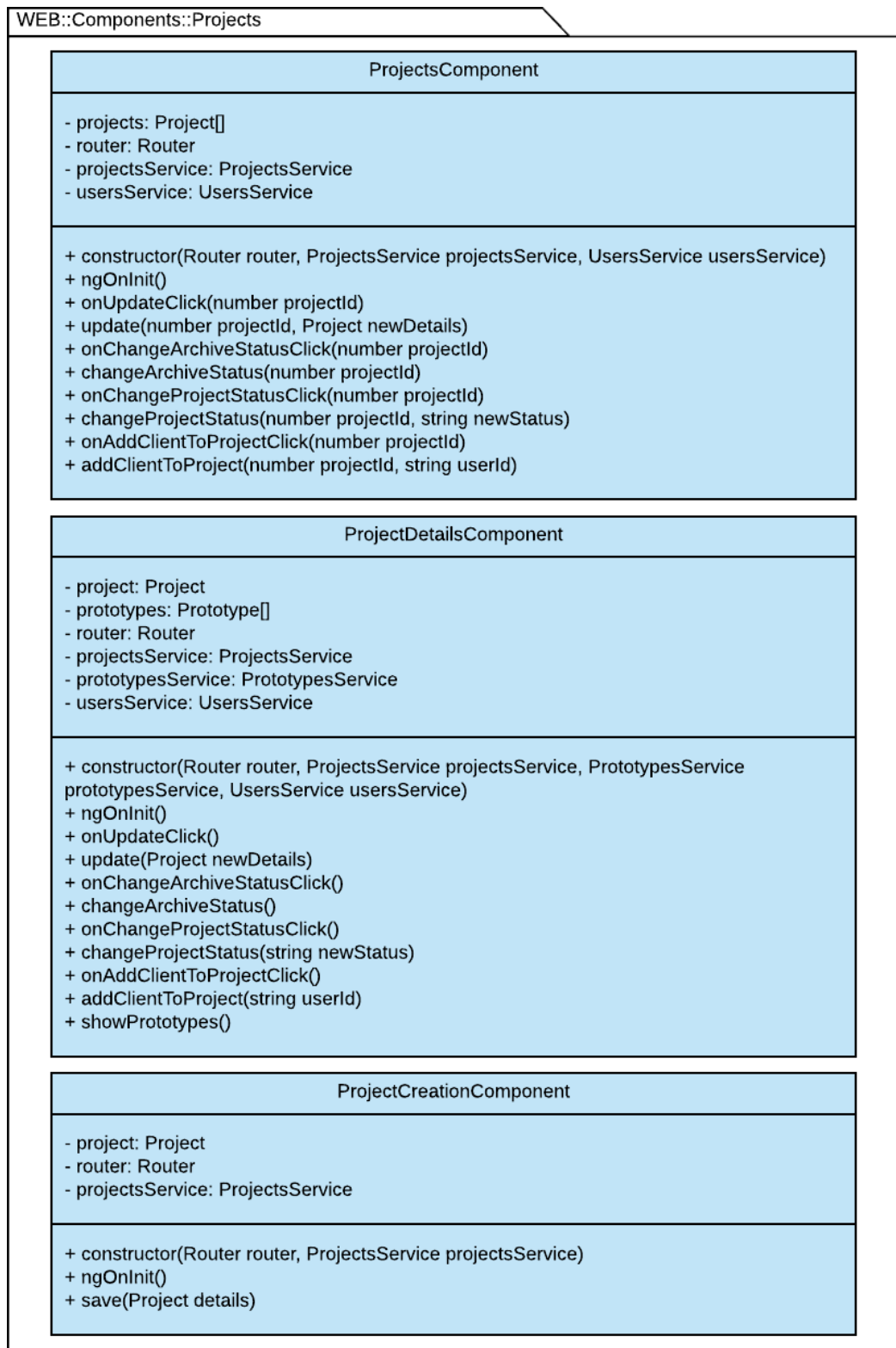
Paketo *WEB::Components::Pictures* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos norint sukurti sistemos vartotojo sąsajos langus ar perpanaudojamus komponentus, skirtus darbui su nuotraukomis. Šis paketas naudoja reikiamas klases iš *WEB::Models* ir *WEB::Services*.



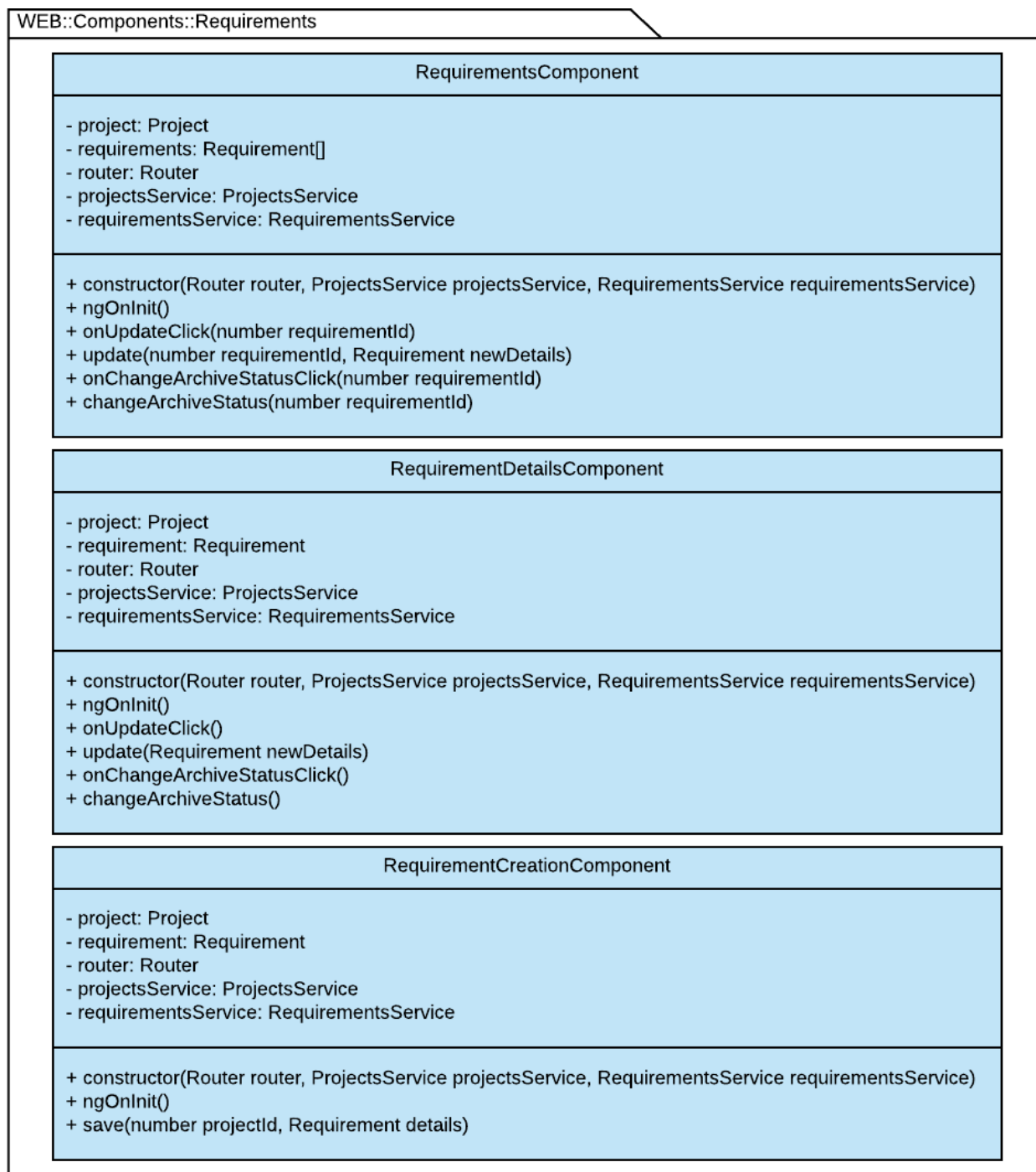
Paketo *WEB::Components::Users* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos norint sukurti sistemos vartotojo sąsajos langus, skirtus darbui su sistemos vartotojais ir jų profiliais. Šis paketas naudoja reikiamas klases iš *WEB::Models* ir *WEB::Services*.



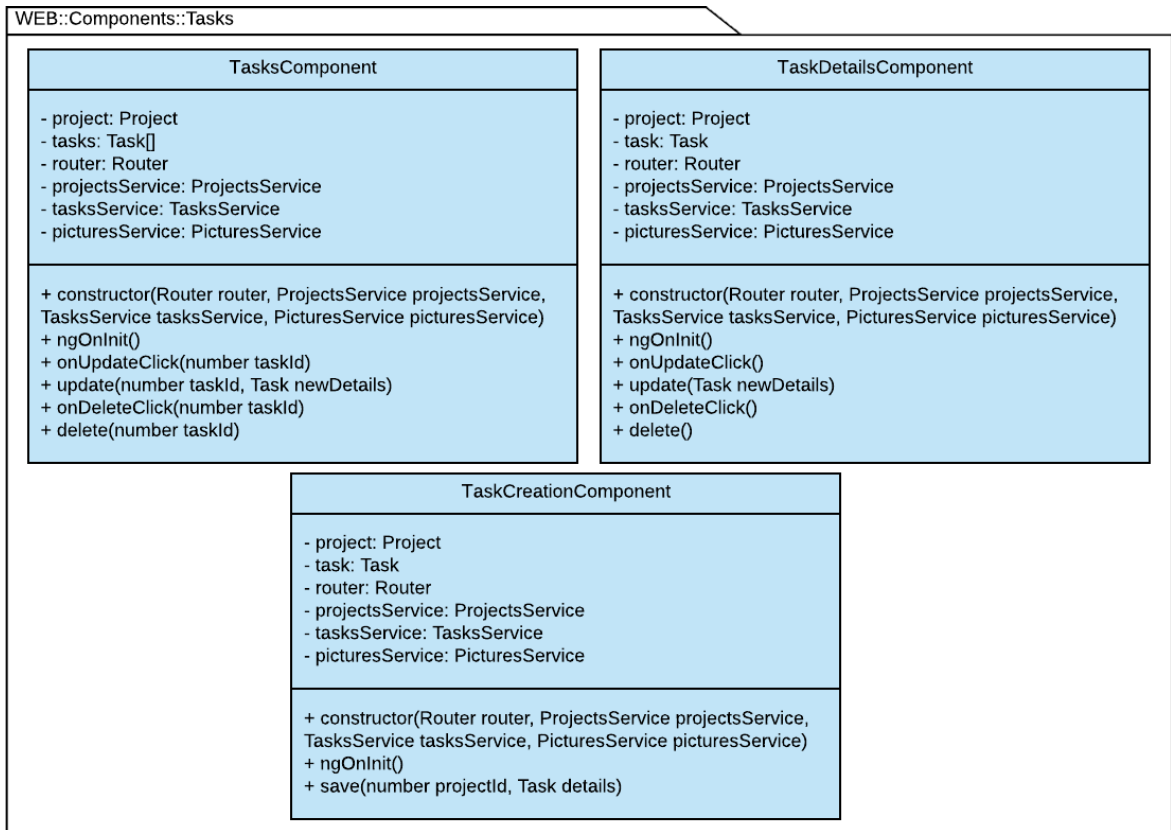
Paketo *WEB::Components::Projects* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos norint sukurti sistemos vartotojo langus, skirtus darbui su projektais. Šis paketas naudoja reikiamas klases iš *WEB::Models* ir *WEB::Services*.



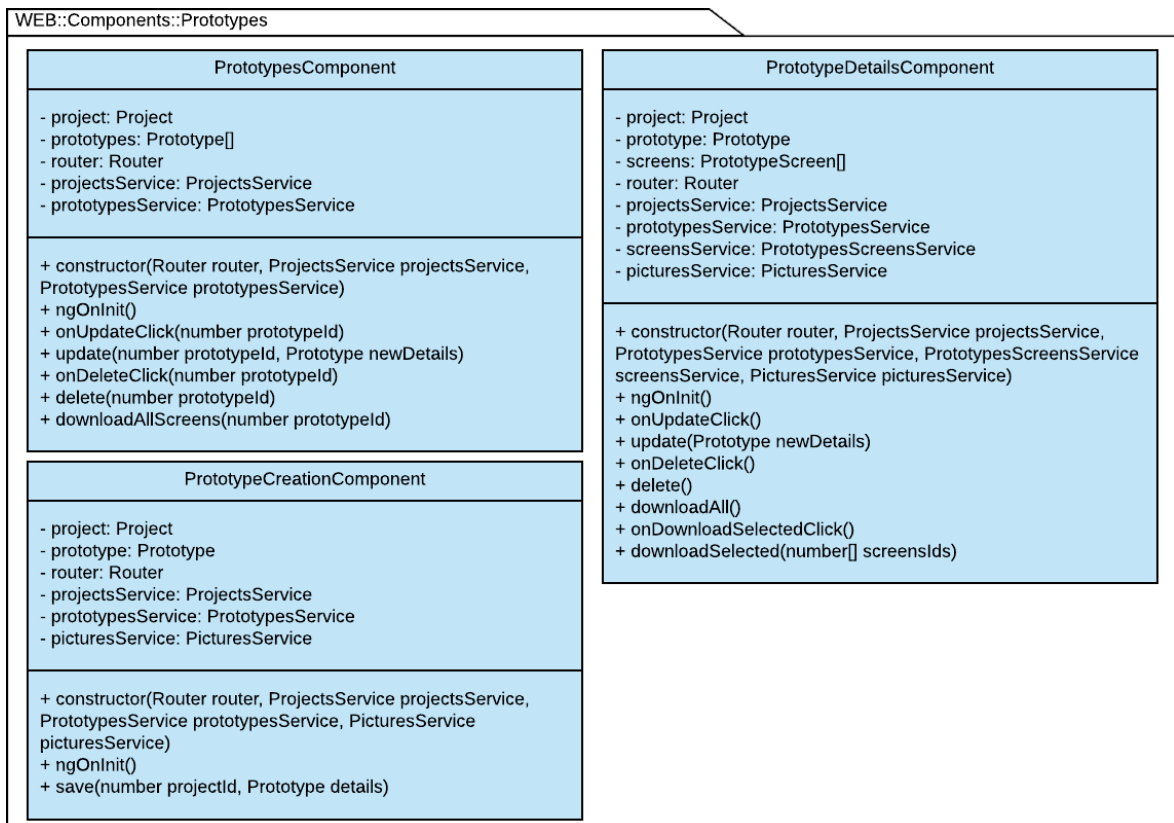
Paketo *WEB::Components::Requirements* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos norint sukurti sistemos vartotojo langus, skirtus darbui su projektų reikalavimais. Šis paketas naudoja reikiamas klases iš *WEB::Models* ir *WEB::Services*.



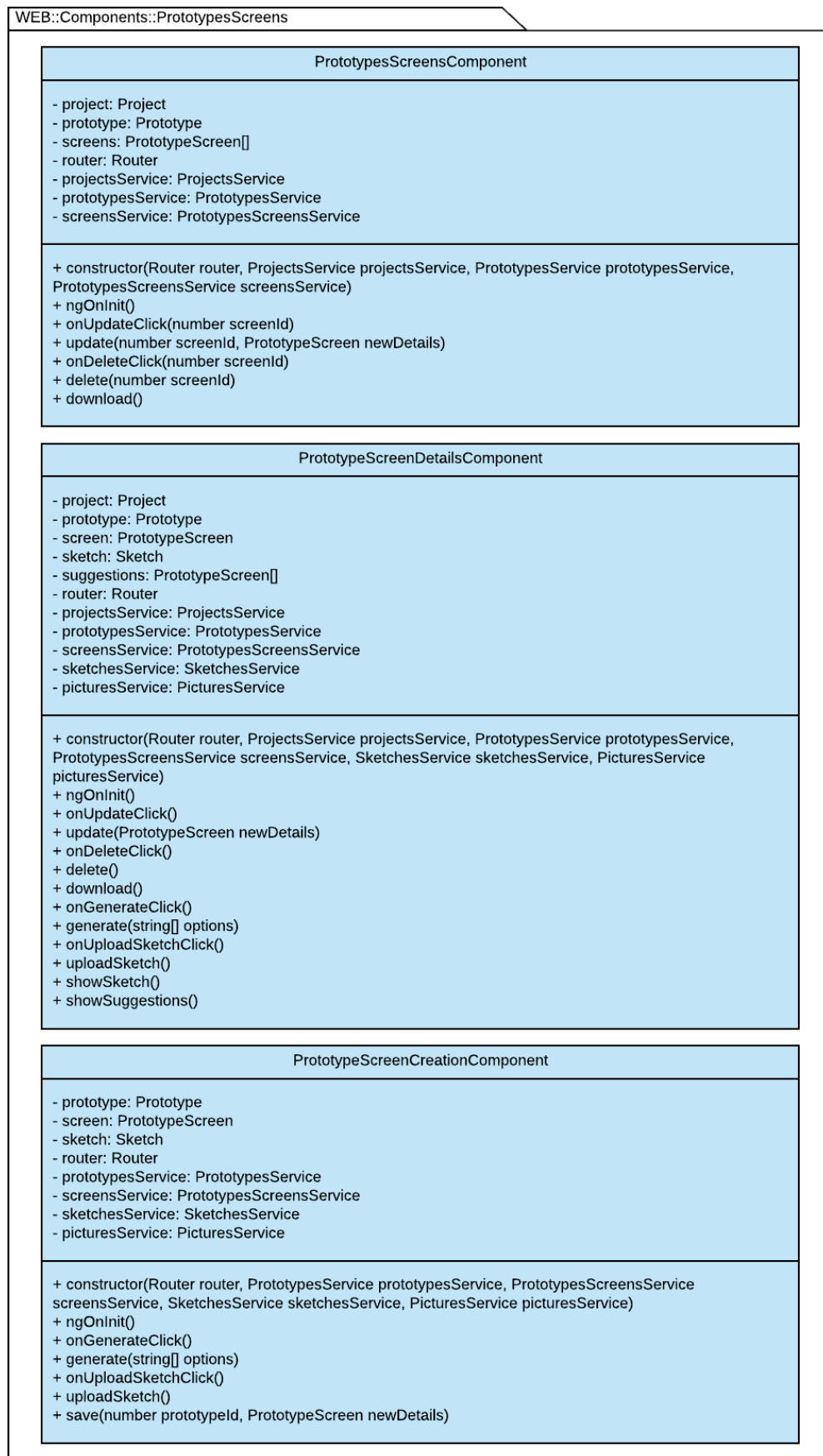
Paketo *WEB::Components::Tasks* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos norint sukurti sistemos vartotojo langus, skirtus darbui su projektų užduotimis. Šis paketas naudoja reikiamas klases iš *WEB::Models* ir *WEB::Services*.



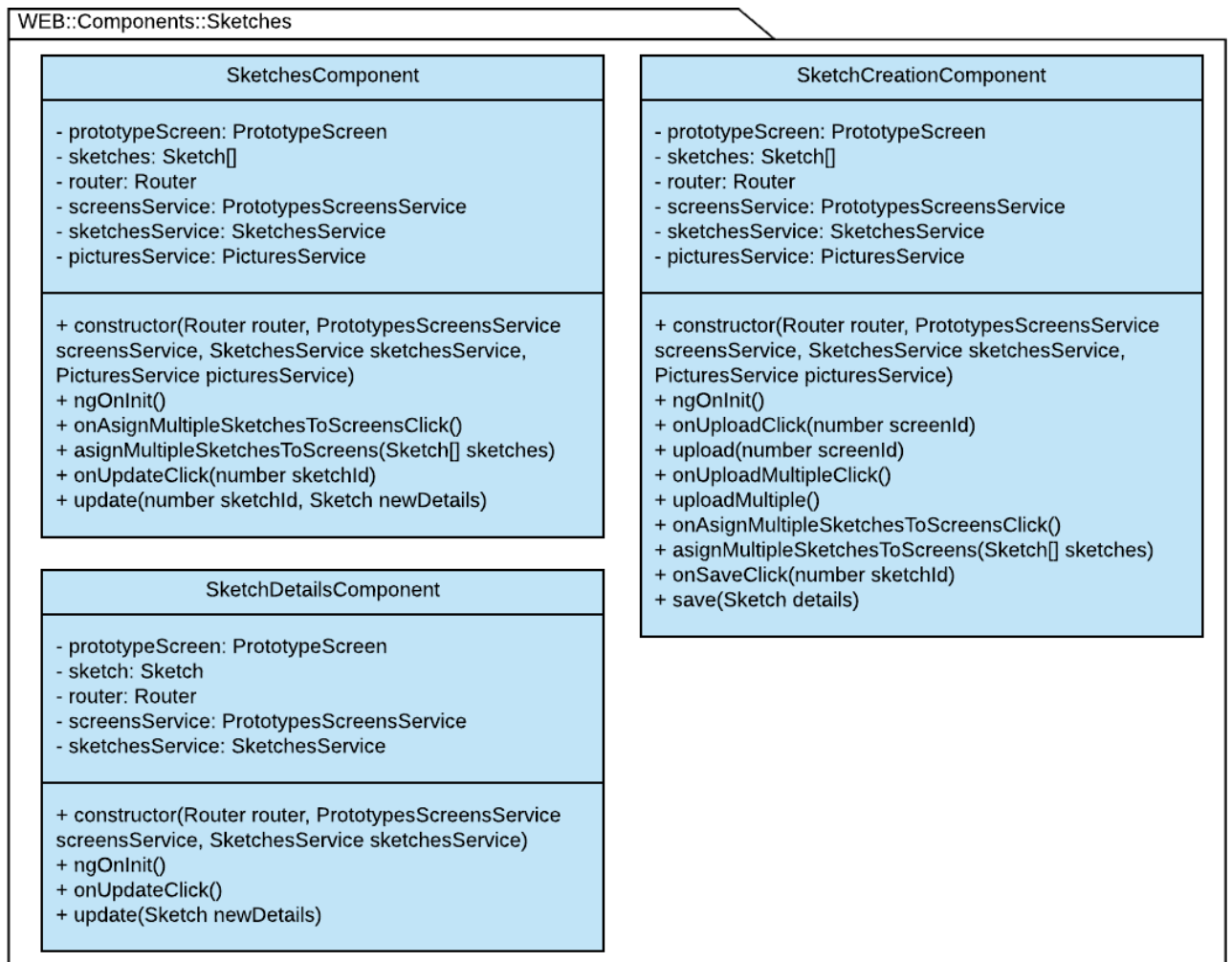
Paketo *WEB::Components::Prototypes* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos norint sukurti sistemos vartotojo langus, skirtus darbui su prototipais. Šis paketas naudoja reikiamas klases iš *WEB::Models* ir *WEB::Services*.



Paketo *WEB::Components::PrototypesScreens* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos norint sukurti sistemos vartotojo langus, skirtus darbui su prototipų langais. Šis paketas naudoja reikiamas klases iš *WEB::Models* ir *WEB::Services*.



Paketo *WEB::Components::Sketches* klasių diagrama matoma žemiau. Šio paketo klasės yra naudojamos norint sukurti sistemos vartotojo langus, skirtus darbui su eskizais. Šis paketas naudoja reikiamas klases iš *WEB::Models* ir *WEB::Services*.



5 priedas. Tinkamumo naudoti testavimo scenarijus

Naršyklėje turi būti atidaryta Google pradinis puslapis ar kitas „neutralus“ puslapis.

Labas, _____. Mano vardas _____. Šiandien kartu išbandysime mūsų kuriamą projektą.

Prieš pradėdamas norėčiau tau paminėti kelis esminius dalykus, kuriuos perskaitysiu, tam kad nieko nepraleisčiau.

Tu jau galbūt įsivaizduoji kodėl čia esi ir ką mes čia veiksime, tačiau aš norėčiau papasakoti apie patį procesą šiek tiek išsamiau. Aš norėčiau kad išmėgintum projektą, kurį mes kuriame, tam kad galėtume išsiaiškinti su kokiomis problemomis gali susidurti būsimi naudotojai. Visas šis seansas turėtų trukti apie pusvalandį.

Pirmiausiai norėčiau pabrėžti, kad mes testuojame projektą ir mūsų idėjas, o ne tave patį. Tu čia niekaip nesuklysi. Iš tikrųjų čia tikriausiai yra vienintelė vieta šiandien, kurioje gali klysti ar strigti nesijaudindamas, kad galbūt kažką blogai darai.

Aš norėčiau paprašyti, kad tuo metu kai bandysi naudotis projektu, tu pasistengtum visas savo mintis reikšti garsiai, t. y.: sakyti į ką žiūri; sakyti ką bandai padaryti ar surasti; sakyti ką tuo metu galvoji. Tai mums gali labai padėti tobulinant projektą.

Taip pat, norėčiau pabrėžti, kad nesijaudintum jog užgausi mūsų jausmus. Mes testuojame projektą norėdami jį tobulinti, todėl mums yra labai svarbi atvira nuomonė.

Jeigu turėsi kokių nors klausimų, tiesiog klausk. Aš galbūt negalėsiu jų atsakyti iškart, nes norėčiau pamatyti kaip tu elgtumeisi jeigu neturėtum ko nors kas tau galėtų padėti. Tačiau jeigu baigus testavimą vis dar turėsi klausimų, aš į juos pasistengsiu atsakyti. Jeigu tau bus reikalinga pertrauka, tiesiog sakyk.

Tikriausiai pastebėjai mikrofoną. Su tavo sutikimu bus įrašinėjamas kompiuterio ekranas ir garsas. Įrašas bus naudojamas tik bandant išsiaiškinti kaip patobulinti šį projektą ir jo daugiau niekas kitas nepamatys apart projektą kuriančios komandos. Ir tai man padės, nes nereikės tiek daug užsirašinėti.

Jeigu sutinki, norėčiau, kad pasirašytum paprastą sutikimo formą. Ji tiesiog sako, kad mes galime įrašinėti kompiuterio ekraną ir mūsų pokalbį, o pats įrašas bus pasiekiamas tik projektą kuriančiai komandai.

Duoti pasirašyti sutikimo formą ekrano ir garso įrašinėjimui.

Kol pasirašinėja, pradėti daryti įrašą.

Gal turi kokių nors klausimų man?

KLAUSIMAI APIE ASMENĮ

Gerai. Prieš pradėdant, norėčiau tavęs paklausti kelių greitų klausimų.

- Kuo šiuo metu užsiimi? Ką veiki per dienas?
- Kiek valandų per savaitę praleidi prie kompiuterio, įskaitant tiek naršymą, tiek įvairių programų naudojimą ar kita? Įskaitant tiek darbe, tiek namie. Nereikia tikslaus skaičiaus, užteks ir pakankamai grubaus.
- Kokiuose tinklalapiuose paprastai lankaisi?
- Kokie tavo mėgstamiausi tinklalapiai?
- Ar naudojiesi sudėtingomis sistemomis, prie kurių būtina prisijungti? Pavyzdžiui, elektroninė bankininkystė.
- Kas tokiose sistemose tau labiausiai patinka?
- O kas labiausiai nepatinka?

Gerai, šaunu. Su klausimais baigėme, galime pradėti testavimą.

Atidaryti pradinį sistemos langą.

TESTAVIMO VEIKLOS / UŽDUOTYS

Pirmiausiai norėsiu paprašyti, kad apžiūrėtum tai ką matai ir pasakotum kas tavo nuomone čia yra: gali pabrėžti detales kurias matai; kieno sistema čia yra; ką čia galėsi atlikti; kam tai yra skirta ir pan. Kol kas tik apsidairyk ir viską nupasakok, tačiau dar ant nieko nespaušk.

Leisti apžiūrinėti pirmą sistemos langą, tačiau ilgiausiai tik kokioms trimis ar keturioms minutėms.

Ačiū. Dabar aš paprašysiu, kad pamėgintum atlikti šiek tiek užduočių, ar kitaip įvardinus veiklą. Aš kiekvieną jų perskaitysiu garsiai bei duosiu atspausdintą popierinę versiją, kad bet kuriuo metu galėtum prisiminti ką turi atlikti.

Taip pat, noriu paminėti, kad projektas vis dar yra kūrimo stadijoje, todėl jei ant ko nors paspaudus nieko neįvyks, tiesiog žinok, kad ten funkcionalumas dar nėra įgyvendintas ir bandyk toliau įgyvendinti užduotis.

Ir žinoma, aš dar kartą paprašysiu, kad visas savo mintis reikštum garsiai, tam kad eitų suprasti kur tau kyla neaiškumai ir kad taip rastume taisytnas vietas.

Duoti dalyviui pirmą veiklos scenarijų ir perskaityti jį garsiai.

Leisti dalyviui tęsti tol kol jis įveikia užduotį, jam nebesigauna ir jis pradeda labai nervintis, arba kol tai nebeteikia jokios naudos.

Tai kartoti su kiekviena užduotimi arba kol baigsis laikas.

Ačiū, tai buvo labai naudinga.

Gal turi kokių nors klausimų, dabar kai jau esame baigę testavimą?

Sustabdyti vaizdo įrašo filmavimą.

Padėkoti už pagalbą ir išlydėti.

6 priedas. Tinkamumo naudoti testavimo užrašai

RESPONDENTAS NR. 1

KLAUSIMAI APIE ASMENĮ

Kuo šiuo metu užsiimi? Ką veiki per dienas?

> *Juokiasi.*

Prie kompo sėdžiu.

Kiek valandų per savaitę praleidi prie kompiuterio, įskaitant naršymą internete, įvairių programų naudojimą, darbą ar kita?

Per savaitę? Čia su darbu?

> *Ieva: Taip.*

Sakykim kokią 60 val.

Kokiuose tinklalapiuose paprastai lankaisi?

Autoplius, skelbiu.lt, ebay, apibendrinant skelbimų tinklalapiuose. Na dar naujienų, bet ten tai retai.

Kokie tavo mėgstamiausi tinklalapiai?

Nėra tokio. Ebay gal nebent. Aliexpress va dar.

Ar naudojiesi sudėtingomis sistemomis, prie kurių būtina prisijungti? Pavyzdžiui, elektroninė bankininkystė.

Taip.

Kas tokiose sistemose tau labiausiai patinka?

Jeigu tenka jungtis kažkur, tai gal tas, kad jautiesi, kad saugiau, nei jei šiaip jungiesi į kokią indo web.

O kas labiausiai nepatinka?

Kad dabar pasikeitę prisijungimai daug kur prie el. parduotuvių. Jeigu bandai apsipirkti, susidedi prekes, eini į krepšelį, tai tave verčia jungtis prie SEB banko, nors pats esu Swedbank. Tai kol ten susijungi... Šitas baisiai erzina.

TESTAVIMO VEIKLOS / UŽDUOTYS

Tikriausiai dar nieko nežinai apie sistemą. Ką darytum norėdama(-as) sužinoti daugiau informacijos (apie tai kas čia yra, kokios galimybės bei kaip naudotis)?

Tai čia yra klaustukas apačioje. Tai tikriausiai ant jo paspausčiau.

> *Nuėjo pažiūrėti į projektus.*

> *Dairosi visur.*

Projektai tai kažkas apčiuopiamo tikriausiai.

Prototipai kažkas ką bandai išgalvoti.

Eskizai ranka kažkas papiešta.

Sakykim jeigu čia būtų susiję su IT, tai čia eskizas projekto eskizas, o prototipas +- vizualizacija, o projektuose tai sukurta konkrečiai tą padaryt, aną padaryt.

Na bet čia mano spėlionės.

Kitas dalykas, gimė idėja dabar, nu kai žmonės ieško kas sukurtų puslapį, nu ir maždaug čia jis papiešia eskizą, ko jam ten reikia. Tai gal čia kažkas iš tą pusę, kad pavyzdžiui galėtų prisiregistruoti freelanceriai, kad galėtų papiešti eskizus, padaryti prototipą ir matytų viską.

Arba gal čia yra galutinis projektas ką čia reikėtų padaryti, ir tada freelanceriai padaro projektą pagal duotus reikalavimus.

> *Ieva: Pabandyk paspausti ant klaustuko, ten paspausti ant "Greita pradžia" ir ten panagrinėti.*

Su šita greita pradžia biškį užtruksiu.

> *Juokiasi.*

Čia labai smulkiai aprašyta, tai čia labai gerai, nes kai reikia kažko tai labai gerai po tokius ieškotis informacijos.

Labai gerai aprašyta praktiškai kiekvienas mygtukas ką jis daro, kaip jį pasiekti ir pan. Ne taip kaip themeforest temoj buvo parašyta "kad va yra mygtukas". O ką jis daro ar kaip jį galima pakeisti, tai teko pačiam išmąstyt.

Tai čia labai gerai kad taip smulkiai viskas yra aprašyta.

> *Paspaudė ant "Naudotojo atmintinė".*

Galvoju ar čia nesikartoja.

Ai čia tokia kaip apžvalga (greita pradžia), o čia (naudotojo atmintinė) žymiai išsamiau ir kiekvienas dalykas.

O čia pradžia (nuoroda dokumentacijos viršuje) neturėtų numesti iš sistemos home page? Man čia atrodo kaip kokio architekto turinio valdymo sistema.

> *Ieva: Architekto, tai to kur pastatus projektuoja ar IT architekto?*

Gal visgi linkstu, kad su statybom čia susiję.

Pabandyk išsiaiškinti ir įvardinti kokią informaciją galima užpildyti apie kiekvieną projektą.

> *Įsijungė projekto kūrimo langą.*

Ai bet tu negali kurti eskizo?

> *Dairosi.*

Tu gali sukurti projektą ir prototipo langą.

O kuo skiriasi prototipo langas nuo prototipo?

Ai tai čia sakykim pagrindinė info, o čia dar keli kiti pages.

> *Pažiūrėjo sukūrimo formas.*

Ai tai čia gan standartinės sukūrimo formos, labai aišku kas jose yra ir t.t.

> *Peržiūri projektą.*

Čia galvoju gal koks nors... ai po biškį aiškėja kas tas yra.

Čia truputį nesuprantu ar čia orientuotas iš freelancerius ar čia yra labiau kaip įmonės koks vidinis puslapis, kur yra sudėti kaip darbai, projektai, asanos sakykim pusseserė. Kaip suprantu čia susiję su IT susiję, svetainių projektavimas, apsirašo viską, kažkas konkretizuoja ir daro.

Čia kaip sakykim ateina nuo gatvės žmogus ir sako man reikia sukurti skelbimų puslapio.

> *Ieva: Tai labiau įmonės vidinis dalykas. Kuriame tarkim bus susirašomi projektai, kiekvieną projektą galima išspręsti įvairiais skirtingais būdais, todėl tam yra prototipai, arba žinoma su jais galima susiskaldyti iš mažesnes dalis, kad projektuoti. Bei langai, na tai kaip pradinis puslapis, apie mus puslapis, vartotojo profilis, mokėjimų langas ar pan.*

Čia kai mes turėjom tą projektą pageidavimų registras, kur viskas susisiunčia iš asana, projektams kurie jau yra sakykim, o čia kad labiau klientai sakyktų norus naujiems projektams.

Apsižvalgyk po projektą ir paaiškink ar supranti skirtumą tarp projektų, prototipų, langų, reikalavimų, užduočių ir kitų duomenų esybių?

Čia kiek matau yra iš įmonės pusės šita turinio valdymo sistema, ne iš kliento. Nes jam tikrai nereikia matyti ten kitų projektų, prototipų ir pan.

> *Vaikšto.*

Ai va kur eskizai, bet negali kurti eskizų. O kur įkelti?

Jeigu naujas projektas tai nematau niekur kur įvesti reikalavimų, užduočių ir pan. Naujas prototipas irgi užduočių nematau. Prototipo langas irgi užduočių nematau kur įvesti.

Kai tu įeini į projektą, tai čia jau yra užduotys. Reikalavimai va yra ir užduotys. Bet projektuose (projekto kūrimo formoj) tai nematau kur čia reikėtų sukurti tai. Gal nebent kokiam sekanciam žingsnyje?

> *Ieva: Gali pamėginti.*

> *Pildo formą.*

> *Sukūrė projektą. Nuėjo prie reikalavimų, užduočių ir pan. bloką, pamatė mygtukus naujų duomenų sukūrimais, paspaudinėjo ant jų.*

Ai supratau, kai sukuri projektą, gali čia pridėti kiek nori. Ir užduočių pridėti kiek nori.

Nu dabar viskas aišku.

> *Persižiūri klausimą iš naujo.*

Skirtumus jo, suprantu. Tik tas žodis "prototipų", nežinau, kažkaip gal čia netinka, šiek tiek maišo mane. Gal neskamba kažkaip. Aš pavyzdžiui ieškočiau kažko kito.

"Prototipas" man atrodo kaip mašiną kurią ir ten prototipuoja, bet dar nepaleistas į masinę gamybę, kur parodo ką gali bet dar nepaleistas į viešumą.

> *Sukuria dar vieną projektą.*

Šiai veiklai įsivaizduok, kad esi projekto vadovas(-ė). Atsirado naujas projektas apie kuri yra žinoma žemiau nurodyta informacija, kaip ją užpildytum?

Reikalavimai:

- Integruoti arba sukurti rezervacijų sistemą, kad būtų galima užsisakyti kambarį.
- Turi būti nuotraukų galerija, leidžianti pamatyti kiekvieną kambarį.
- Vartotojai turi galėti matyti savo profilį, aktyvias ar pasibaigusias rezervacijas, mokėjimus ir kitą susijusią informaciją.
- Būtina reprezentatyvi dalis, kurioje būtų informacija apie mus, kontaktai, kur randamės ir t.t.

Užsakovas:

- Jonas Jonauskas.

Prie projekto tinkamiausi dirbti darbuotojai:

- Gintarė Gintarytė.
- Tomas Tomauskas.

Tai čia pabandyt pakurt toki?

> *Ieva: Taip.*

> *Kuria projektą. Prideda vartotojus. Kuria reikalavimą.*

> *Atsidarė reikalavimas, susimaišė kaip grįžti į projektą, ieškojo per projektus.*

> *Sukūrė dar vieną reikalavimą.*

> *Ieva: Šiaip yra greitesnis būdas grįžti.*

> *Apsidairė, rado breadcrumb, paspaudė ant jo.*

Na viskas kaip ir aišku ir tvarkingai šitoje vietoje.

Tik reikalavimai galėtų būti labiau suspausti. Toks vaizdas, kad daug bereikalingos vietos užimta, gal juos stulpelias sudėti, aprašymo numesti. Kad taupyti vietą, kad mažiau scrollinti reikėtų per visą šitą puslapį (projektą).

Įsivaizduok, kad esi dizaineris(-ė), kuris projektuos visą vartotojo sąsają. Kokius prototipus bei langus šiame projekte susikurtum? Pabandyk juos sukurti. Jeigu prireiks, eskizų nuotraukos yra atidarytame aplanke.

Kaip galėtum sugeneruoti ir peržiūrėti sistemos darbo rezultatus?

Kas ta vartotojo sąsaja?

> *Paaiškinau.*

Ai tai čia eina eilės tvarka? Susikuri projektus, tuomet prototipus ir tik tada eskizus.

> *Kuria prototipą.*

- > Kilo klausimų, pasikalbėjom, nes nebuvo aišku kaip skaidyti prototipus ir langus. Bet neaiškinau pilnai, kad pamatyt kaip elgtųsi jei nebūtų kas padėtų.
- > Dar pagalvojo kaip jam susidėliot viską.
- > Ieva: Gali įeiti į pirmą projektą ir ten apsidairyti.
- > Nuėjo, apsidairė. Pasivaikščiojo po projekto prototipus ir langus juose.

Aaaai, viskas aišku.

Tai gal aš imu tarkim pirmą reikalavimą ir pagal jį tada tik padarau.

- > Kuria prototipą - kambario rezervacijos dizainas.
- > Kuria langą - žmogaus duomenų įvedimo vaizdas.
- > Dairosi po sukurtą langą.

Pasiūlymai, kodėl generuoti? Ką čia generuoti?

- > BUG: kai kuriant langą nėra pridėto eskizo. Generavimo mygtukas be teksto. Jį paspaudus įvyksta negerai dalykai.
- > Kuria dar vieną langą - kalendoriaus rezervacija.

Negaliu redaguoti kas sukurta?

- > Ieva: Ne, dar kol kas negalima.

Na tai dizaineris turi čia sudėti viską, tada užsakovas pažiūri ar gerai viskas.

- > Priminiau užduoties antrą dalį.
- > Pabandė pageneruoti prototipą (lange, iš eskizo į html).
- > Paspaudė ant parsisiųsti html.
- > Šiek tiek nustebęs ir šiek tiek suglumęs.

Net neturiu ką apibūdint šitoje vietoje.

Čia gal neturėtų būti belenkokios raidės, o gal kas nors susiję su tuo eskizu.

Žinok neišivaizduoju kas čia.

- > Ieva: O supratai kas čia įvyko?

Kaip galvoju tai pirmas variantas. Gal čia šitą prototipą pakonvertavo į html, kad būtų galima nusiųsti linką.

Antras variantas. Čia gal sugeneravo šito prototipo vaizdą.

- > Nuėjo į prototipo peržiūrą, grįžo prie lango.

Ai tai jei generuoti čia yra, tai tuomet šitą langą ir generavo. Tada gali nusiųsti linką, pasakytų taip ar ne taip.

Čia greičiausiai klientas neturėtų prisijungimo prie šitos sistemos, greičiausiai čia nieko nematytų. O galėtų tik ką reikia gauti per link.

- > Grįžo į prototipo peržiūrą, kur matosi visi jame esantys langai.

O čia kur prototipe generuoti nėra?

Ai tai čia supratau, kiekvieną po vieną supieši, sugeneruoti ir žiūri kas gerai kas ne.

Įsivaizduok, kad esi projekto vadovas(-ė). Pabandyk išsiaiškinti prie kurių projektų dabar dirba Tomas Tomauskas.

- > Ėjo į vartotojai, susirado Tomas Tomauskas, paspaudė, pascrollino.
- Easy.
- > Juokiasi.

Pabandyk išsiaiškinti kurie projektai buvo atšaukti.

- > Eina į projektus, pasirenka lentelę, bando rikiuoti pagal būsenos stulpelį.
- Va.
- > Dar pasidairė prie bloko viršaus, kur yra atvaizdavimo būdai ir filtravimas (paieška).
- Dar būtų gerai ir čia kur nors viršuje filtras.

Pabandyk išsiaiškinti kaip sužinotum kurios užduotys iš bet kurio projekto vis dar nėra įgyvendintos.

- > Nuėjo į užduočių langą, pasirinko žiūrėti pagal lentelę, apsidairė prie būsenos stulpelio.
- O tada klausimas, tai čia yra visos užduotys kiek tik sukurta per visur?
- > Ieva: Taip.

RESPONDENTAS NR. 2

KLAUSIMAI APIE ASMENĮ

Kuo šiuo metu užsiimi? Ką veiki per dienas?

Dirbu, kuriu web dizainą, administruoju wordpressą, ir dizaino užduotis atlieku.

Kiek valandų per savaitę praleidi prie kompiuterio, įskaitant naršymą internete, įvairių programų naudojimą, darbą ar kita?

Nuo 7 val. iki 23 vakaro. 16 valandų per dieną.

> *Ieva: O tu kažkada nesakei, kad eini miegot 21 val.?*

Anksčiau eidavau miegot 21, bet dabar išsiderinau.

> *Susijuokė.*

Kokiuose tinklalapiuose paprastai lankaisi?

Facebook, youtube, skillshare mokslams, instagram, gmail, spotify. Yra daugiau, bet čia pagrindinės.

Kokie tavo mėgstamiausi tinklalapiai?

Kas keisčiausia, kad nei vieno nemėgstu kuriuose lankaisi. Sakykim palikim instagram. Facebook pagal nuotaiką, nes kartais pataikai, kartais š*** klanas. Skillshare visada palieka gerą nuotaiką, bet nesilankau taip dažnai kaip kituose.

Ar naudojiesi sudėtingomis sistemomis, prie kurių būtina prisijungti? Pavyzdžiui, elektroninė bankininkystė.

Jo, naudoju kur reikia sodra, apmokėjimai, swedbank, mokesčių mokėjimai, tai su visais tais dalykais tenka susidurti.

Kas tokiose sistemose tau labiausiai patinka?

Labiausiai patinka kai pažingsniui yra padaryti dalykai. Pavyzdžiui, patiko kaip tele2 dabar yra padaryti apmokėjimai, dideli mygtukai, aiškūs žingsniai ir t.t.

Patinka aiškumas, pažingsniui, kai nėra kažkokių siurprizų, aišku kas laukia sekančiame žingsnyje, nėra apkrauta.

O kas labiausiai nepatinka?

Kur viskas randasi visiškai belekur, neaišku kur ko ieškoti, neaišku kas sekančiam žingsnyje bus, neaiškūs pavadinimai ir užrašai, nežinai kokius pasirinkimus gausi paspaudus ant kažkokio mygtuko, galbūt kartais net nesaugumo jausmas nes interface toks nefainas atrodo, kur nėra aišku, kur labai daug visko vienam puslapyje ar žingsnyje. Ar kas labiausiai erzina, kai tu bandai prisijungti, bet tave nebūtinai prijungia prie kur tau reikia.

TESTAVIMO VEIKLOS / UŽDUOTYS

Tikriausiai dar nieko nežinai apie sistemą. Ką darytum norėdama(-as) sužinoti daugiau informacijos (apie tai kas čia yra, kokios galimybės bei kaip naudotis)?

Na tai čia iškart matau yra mygtukas "Kaip naudotis?". Spaudžiu ant jo.

> *Iškart pasiekė dokumentaciją.*

> *Naršo dokumentaciją, pradinis langas suglumino, nes nėra aišku, kad galima rasti daugiau info*

> Grįžo į sistemą, naršo po ją.

> Grįžo prie dokumentacijos.

> Nuėjo į vartotojo atmintinę.

Paskaitau, bet norėčiau paprastesnės informacijos paieškoti.

Yra greita pradžia, paspaudžiu ant greitos pradžios.

O čia daugiau dalykų. Čia yra ko galbūt man reikėtų. Tai paskaitysiu.

Apžvalga, va čia tas kas į temą atrodo.

> Žiūri apžvalgos skyrių.

Čia nežinau kiek čia svarbu, bet pažiūrėjus šitas ikonas (sistemos screenshotas - viršutinės juostos papildomi veiksmai), pažiūrėjau viršuje dešinėje, tikėdamasi jas rasti, nors sąmoningai žinojau, kad čia dokumentacija, ne sistema.

> Pažiūri į scroll bar.

Iš pradžių neatkreipiau dėmesio, kad čia labai daug visko yra. Sakykim pavargau skaityt, norėčiau trumpesnės versijos.

Kaip naudotis radau, bet kas čia yra man vis dar trūksta skilties "apie". Jeigu manęs niekas neforcinėtų, tai pati tikriausiai daugiau informacijos nebeieškočiau.

> Žiūri dokumentacijos pradinį langą.

Aš čia tikėčiau, kad būtų kokia trumpa informacija apie tai kas čia yra. Nes atrodo, kad čia šita pradžia neišbaigta ir nežinai, kad čia daugiau galima rasti. Bet aišku jei čia būtų surašyta bent pradinė informacija, tai būtų aišku kas per projektas, ką jame galima daryti ir tada būtų galima pasigilinti ko konkrečiau reikia.

Pabandyk išsiaiškinti ir įvardinti kokią informaciją galima užpildyti apie kiekvieną projektą.

Kadangi manęs čia niekas neprašo nieko pildyti, tai...

> Eina į dokumentaciją, spaudžia "Greita pradžia".

> Rado skyrių apie bendrą duomenų struktūrą (projektai, prototipai, langai). Paskaito.

Ok, projektas gali turėti užsakovus ir darbuotojus, projekte gali būti prototipai, reikalavimai ir užduotys.

Dar galiu pasižiūrėt projekto kūrimą. Bet pradinė informacija jau gauta.

> Žiūri apie projekto kūrimą.

Nieko neskaitau, visą reikiamą informaciją matau iš paveiksluko ir paryškintų tekstų.

> Rado prototipo kūrimo formą.

Čia viskas yra, yra pavadinimas aprašymas ir t.t.

Apsižvalgyk po projektą ir paaiškink ar supranti skirtumą tarp projektų, prototipų, langų, reikalavimų, užduočių ir kitų duomenų esybių?

Lieku tam pačiam lange (dokumentacijos), grįžtu prie bendros struktūros.

Ok viską perskaičiau. Atrodo parašyta sudėtinga, bet kai prisiminiau kad čia apie svetaines, tai čia iš tikrųjų labai paprasta viskas.

> Apibūdino kaip veikia prototipai, langai ir pan.

Dabar pagalvojau, kad norėčiau pakurti projektą. Bet dar nenoriu, kad neprikurt nesąmonių, tai vėl pažiūrėsiu dokumentaciją. Kad saugiai su viskuo susipažinčiau.

> Perėjo į sistemą, žvalgosi.

Prie projektų iš karto matau mėlyna, pilka, būsenas projektų.

Paspaudžiu ant šito projekto, čia yra pagrindinė informacija apie patį projektą. Viskas ką perskaičiau dokumentacijoje čia yra, atitinka realybę, dėl ko jaučiuosi fainai.

Toliau yra prototipai, susiję vartotojai, reikalavimai, yra du reikalavimai, ir užduotys, nėra sukurtų užduočių.

Grįžtu į viršų. Pasiskaitau iš naujo užduotį (testavimo veiklą).

Nu dar pažiūriu prototipą.

Labai smagu kad matau kelią (breadcrumb), kad žinau kur aš esu.

Tada noriu grįžti, spaudžiu čia (ant projekto pavadinimo), grįžau, labai fainai.

Norėčiau prie projekto tokios pačios rodyklės, kad grįžti prie visų projektų.

Tai supratau skirtumus, dokumentacija sutapo.

Projektas sakykim bendras folderis, o jame yra prototipai, reikalavimai, užduotys, vartotojai.

Prototipų gali būti keli, kaip ir langų, reikalavimų ir užduočių.

Reikalavimai kiek suprantu ką projektas turi išpildyt.

Užduotys, kad kažkas turi būti padaryta.
Šiek tiek abejoju skirtume tarp reikalavimų ir užduočių.
Galvoj bandau įsivaizduot kad reikalavimas yra stambesnis dalykas nei užduotis.

Šiai veiklai įsivaizduok, kad esi projekto vadovas(-ė). Atsirado naujas projektas apie kurį yra žinoma žemiau nurodyta informacija, kaip ją užpildytum?

Reikalavimai:

- Integruoti arba sukurti rezervacijų sistemą, kad būtų galima užsisakyti kambarį.
- Turi būti nuotraukų galerija, leidžianti pamatyti kiekvieną kambarį.
- Vartotojai turi galėti matyti savo profilį, aktyvias ar pasibaigusias rezervacijas, mokėjimus ir kitą susijusią informaciją.
- Būtina reprezentatyvi dalis, kurioje būtų informacija apie mus, kontaktai, kur randamės ir t.t.

Užsakovas:

- Jonas Jonauskas.

Prie projekto tinkamiausi dirbti darbuotojai:

- Gintarė Gintarytė.
- Tomas Tomauskas.

Jeigu atsirado naujas projektas, tai kuriu naują projektą (iš top bar). Sudvejoju ar nesukurs projekto projekte, bet vistiek pabandau.

Okei, tai projektas bus rezervacijų sistema.

Aprašymas, tai tikiuosi galiu praleist, nes kol kas nenoriu aprašinėti. Arba gal užrašau kažką.

Kadangi datų neturiu, tai susidėsiu savo nuožiūra.

Pridedu paveiksluką.

> *BUG: kažkas nenusiredirectino normaliai.*

> *Nuėjau į projektą, nes sistema pati turėjo nukreipti į projektą.*

Aiškiausiai bus su reikalavimais, todėl pradėdam su jais.

Pradedu galvoti, kad čia po vieną reikalavimą reikia suvesti.

> *Sukūrė reikalavimą. Patikrino informaciją. Sudvejojo kaip grįžti.*

Paspaudžiau ant reikalavimų. Čia daug reikalavimų.

Ieškau savo projekto. Patikrinu ar yra reikalavimų, yra vienas reikalavimas. Pildau dar vieną reikalavimą.

> *Sukūrė.*

> *Grįžo į projektą su breadcrumb.*

> *Pildo kitus reikalavimus.*

Ok sukūriau dalykus, grįžtam į projektą. Žiūrim toliau ko mums reikia.

Grįžtam prie vartotojai, dar pažiūrime ar tikrai kuriasi kaip vartotojai, o ne atskirai.

Kuriam naują vartotoją.

> *Pamatė, kad vartotojo susiejimo formoje yra pasirinkimas darbuotojui ir užsakovui, apsidžiaugė.*

Nice radau užsakovą. Oho kiek čia vartotojų jau prikurta. Matom Joną Jonauską.

Pasirenku.

Pridėdam darbuotojus.

Ir tada pasitikrinam.

> *Apžiūri projektą.*

Turim tris dalyvius, Tada yra reikalavimai.

Ir tada laiminga nueinu į pradžių. Darbas baigtas, galiu ilsėtis.

Įsivaizduok, kad esi dizaineris(-ė), kuris projektuos visą vartotojo sąsają. Kokius prototipus bei langus šiame projekte susikurtum? Pabandyk juos sukurti. Jeigu prireiks, eskizų nuotraukos yra atidarytame aplanke.

Kaip galėtum sugeneruoti ir peržiūrėti sistemos darbo rezultatus?

Tai aš kaip dizainerė ateinu į projektą

> *Paspaudė ant pačio pirmo iš "Naujausi projektai" per home page.*

Matau iškart "naujas prototipas", galiu kurt, bet pirma pažiūrėsiu reikalavimus.

Tada mes turim turėt daug puslapių realiai.
Tai aš tai norėčiau prototipo kūrimą atsidaryti naujame tab'e, kad galėčiau matyti reikalavimus.
Pradėsiu nuo paprastesnio, tai bus apie mus.
> *Renkasi foto.*
Ne, pasirinksiu kitą foto.
Prie detalizavimo lygio pažiūriu klaustuką.
> *Perskaityto klaustuko pateiktą informaciją.*
Gerai, tai detalizavimas aišku, pasirenku.
Paspaudžiu sukurti ir dabar galvoju ar sukūrė ar ne. Na lygtais yra.
Nėra prototipo langų, tai ką aš tada čia sukūriau? Grįžtu į projektą kad pasižiūrėt ką čia sukūriau.
Tada einu į dokumentaciją, nueinu prie prototipų kūrimas ir peržiūra.
Čia va prototipo kūrimo forma, yra informacijos, apie pavadinimą, viso lango sprendimas ar pan, ai va, supratau.
> *Norėjo atnaujinti ar ištrinti, bet išpėjau kad nėra.*
Kuriu naują prototipą.
Kurių naujus langus, matau eskizo nuotrauka, dar paskaitau, dabar mano teorija pasitvirtina kad darau viską gerai.
Čia matau mygtuką "generuoti", paspaudžiu ant jo, rašo laukiama, laukiu.
> *Laukia.*
> *Susigeneravo, atsirado du nauji mygtukai, juos apžiūri iš naujo.*
Dar "generuoti", ką čia generuoti?
Paspaudžiu ant "parsisiųsti html".
Pasižiūriu ar viskas ok.
Lyg ir viskas ok.
Ok, darbą padariau, galiu namo.
> *Dar pasiūlau pažiūrėti pirmą projektą ir jame esančią informaciją.*
> *Nuėjo, dairosi.*
Dar noriu pažiūrėti užduotis, nes dar nebuvo susidūrusi.
Eik tu kiek čia daug info (bendrame užduočių lange prie kiekvienos užduoties atskirai).
> *Įeina į užduotį.*
Čia info, čia prisegtos nuotraukos. Ai tai čia tiek ir yra.

Įsivaizduok, kad esi projekto vadovas(-ė). Pabandyk išsiaiškinti prie kurių projektų dabar dirba Tomas Tomauskas.

Žiūriu vartotojų dalį.
Nueinu į visi vartotojai.
O Tomas Tomauskas, jis yra dizaineris, spaudžiu ant jo.
Trys projektai.

Pabandyk išsiaiškinti kurie projektai buvo atšaukti.

O, įdomu.
Kažkodėl noriu eiti į projektus.
Tada žiūriu filtrus, bet tada galvoju...
> *Kol kas nėra filtravimo mygtuko. Tai greit paieško kitur / kitaip.*
Pasižiūriu lentelę, bandau rūšiuoti pagal būsenos stulpelį.

Pabandyk išsiaiškinti kaip sužinotum kurios užduotys iš bet kurio projekto vis dar nėra įgyvendintos.

Užduotys, tai man reik užduočių iš visų projektų. Galvoju kur čia rast, o iškart matau meniu.
Šiaip ieškočiau kaip praeitoje veikloje.
Tada kartoju tą patį, einu į lentelę.
Bet nerandu informacijos apie tai kuriame projekte yra ta užduotis. Norėtusi žinoti kokia užduotis kurio projekto.