

VILNAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Ligoninės registracijos sistema
Hospital registration system

Projektinis darbas

Atliko: 2 kurso 4 grupės studentai

Tomas Narevičius (parašas)

Ignas Bujevičius (parašas)

Mykolas Skrodenis (parašas)

Darbo vadovas: Lekt. Artūras Jankus (parašas)

Vilnius – 2019

Turinys

ĮVADAS	3
1. UŽDUOTYS IR JŲ VYKDYMO SCENARIJAI.....	4
1.1. UŽDUOTYS.....	4
1.2. VYKDYMO SCENARIJAI	5
1.2.1. Vartotojo registracija pas daktarą	6
1.2.2. Administratoriaus užduotys	7
2. STRUKTŪRINIS PROGRAMŲ SISTEMOS MODELIS.....	8
2.1. DAKTARO VIZITO REGISTRACIJA	9
2.2. VARTOTOJO REZERVUOTŲ DAKTARO TALONŲ PERŽIŪRA	10
2.3. DAKTARO INFORMACIJOS PERŽIŪRA	12
3. DINAMINIS PROGRAMŲ SISTEMOS MODELIS	13
3.1. DAKTARO VIZITO REGISTRACIJA	13
3.2. INFORMACINIS SISTEMOS PRIŽIŪRĖJIMAS	14
4. PROGRAMŲ SISTEMOS KOMPONENTAI.....	15
5. KOMPONENTŲ ĮSIDĖSTYMAS TINKLE	17
TERMINŲ ŽODYNAS.....	18
LITERATŪROS SĄRAŠAS	19
PRIEDAI	20
1. SAUGUMO REIKALAVIMAI	20

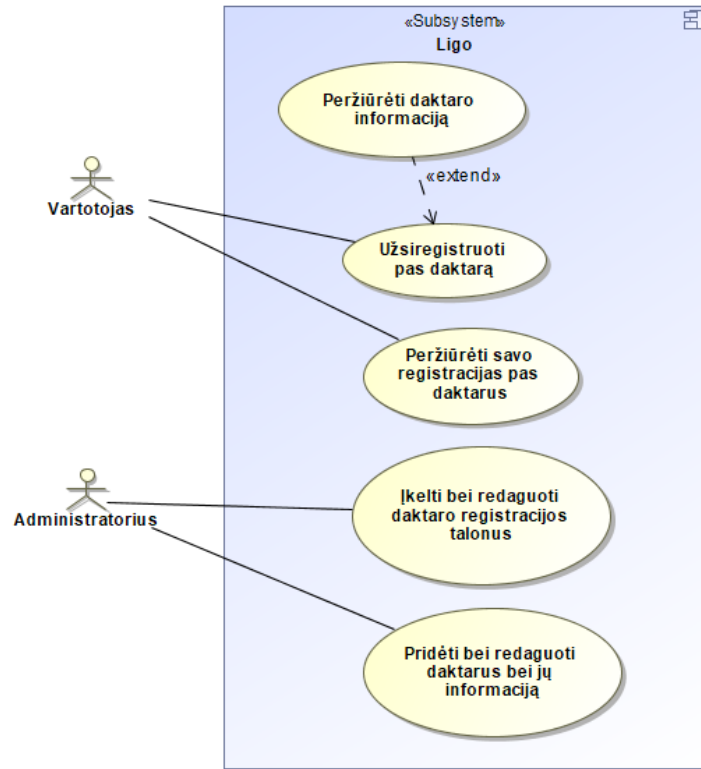
Įvadas

Šis projektinis darbas yra skirtas pavaizduoti ligoninės registracijos sistemos projektą remiantis „UML 4+1“ architektūriniu modeliu, sukurtu remiantis prieš tai sukurtu sistemos prototipu. Sistema autorių pavadinta „Ligo“, remiantis jos prototipo pavadinimu.

1. Užduotys ir jų vykdymo scenarijai

1.1. Užduotys

1 diagrama. „Ligo“ UC (use case) diagrama



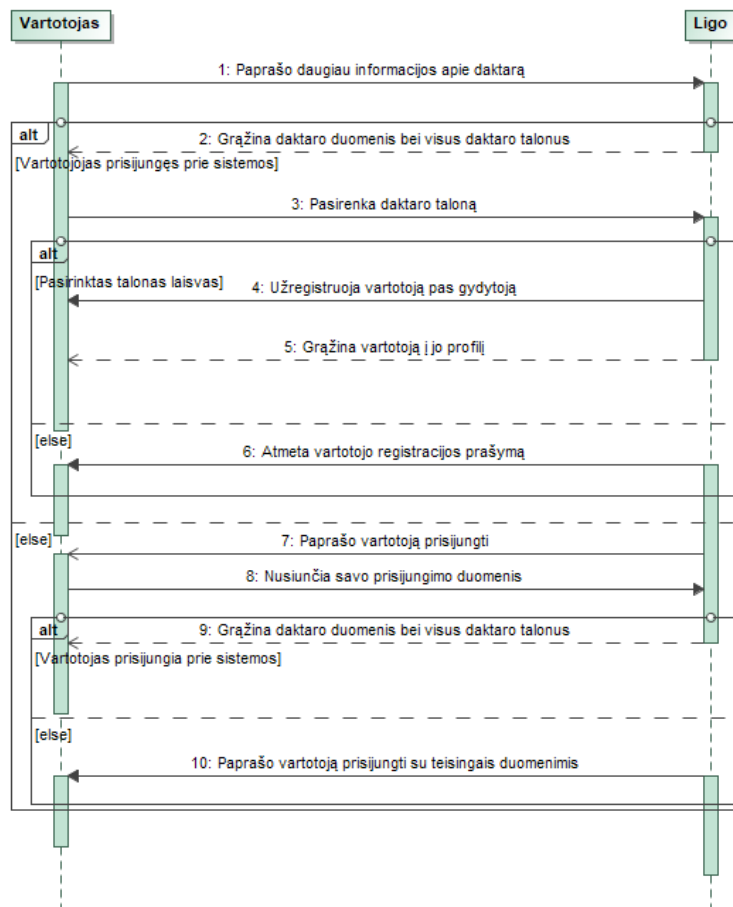
Pagrindiniai aktoriai yra vartotojas, kuris atlieka sistemos naudotojo vaidmenį bei administratorius, kuris atlieka sistemos prižiūrėtojo vaidmenį. Vartotojas, registruodamasis pas daktarą, tuo pačiu metu gauna ir jo išsamesnę informaciją. Norint sužinoti išsamesnę daktaro informaciją, registruotis pas jį nereikia.

1.2. Vykdymo scenarijai

Šiame projekte visi pirmoje diagramoje pavaizduoti tikslai paaiškinami per du vykdymo scenarijus. Daktaro informacija gaunama registruojantis pas jį. Visų vartotojo registracijų peržiūra įkomponuota į registraciją pas daktarą bei trivialiai apibūdinama turint registracijos sekų diagramą. Skirtumas tarp administratoriaus sąveikavimo su registracijomis bei daktarais taip pat trivialus, todėl vaizduojamas, kaip vienas bendras scenarijus.

1.2.1. Vartotojo registracija pas daktarą

2 diagrama. Vartotojo registracijos pas gydytoją sekų diagrama

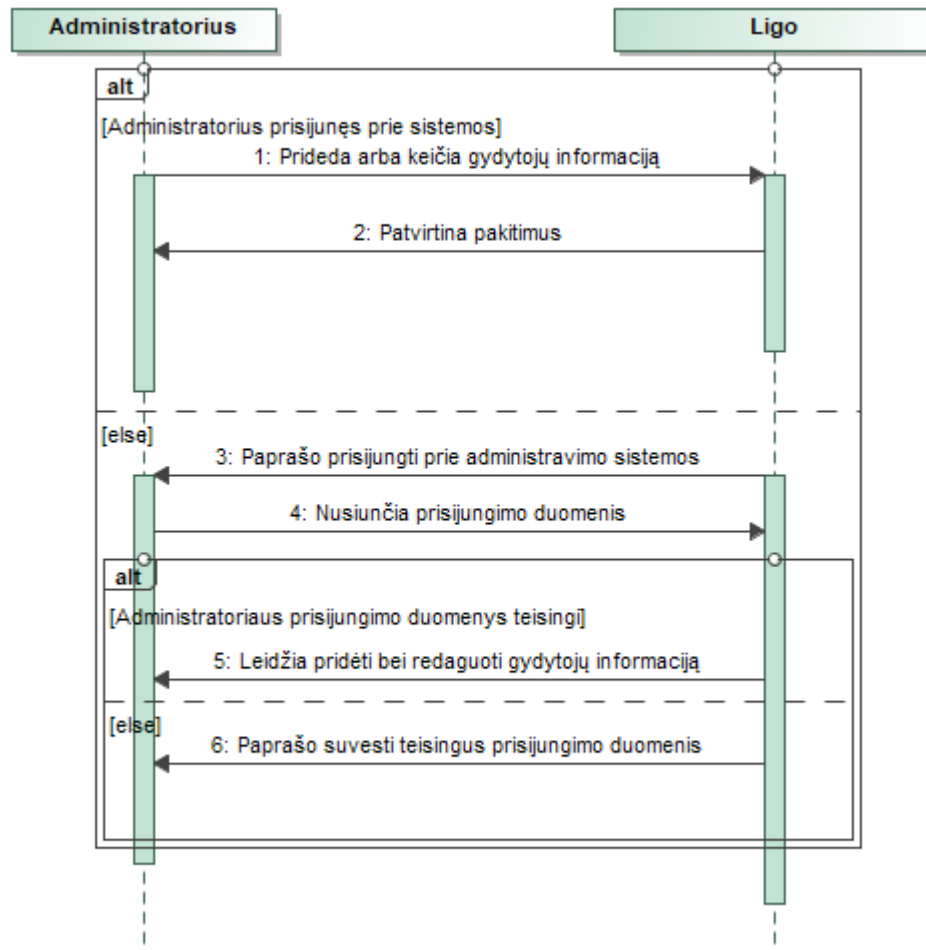


Norint peržiūrėti daktaro talonus bei išsamesnę informaciją, vartotojas turi būti užsiregistravęs „Ligo“ sistemoje. Šiame scenarijuje gali iškilti kelios problemos:

1. Prarandamas ryšys su prisijungimo duomenų baze. Šiuo atveju daktaro duomenys taip pat tampa nepasiekiami, kaip ir privati vartotojo informacija.
2. Vartotojas bando užregistruoti daktaro taloną kitam vartotojui. Šiuo atveju sistema įgyvendintų vartotojo tapatybės patikrinimą bei atmestų vartotojo prašymą.

1.2.2. Administratoriaus užduotys

3 diagrama. Administratoriaus užduočių sekų diagrama



Šioje diagramoje pavaizduotas naujo daktaro pridėjimo bei esamo redagavimo sekų diagrama. Naujų registracijos talonų sukūrimo bei esamų redagavimo vykdymo scenarijus analogiškas viršuje pavaizduotai diagramai, todėl šiame dokumente nevaizduojamas. Skaitytojų gali kilti klausimas, ar galima prisijungti bei redaguoti daktarų informaciją vartotojams? Vartotojas neturi teisės prisijungti prie administratoriaus sistemos.

2. Struktūrinis programų sistemos modelis

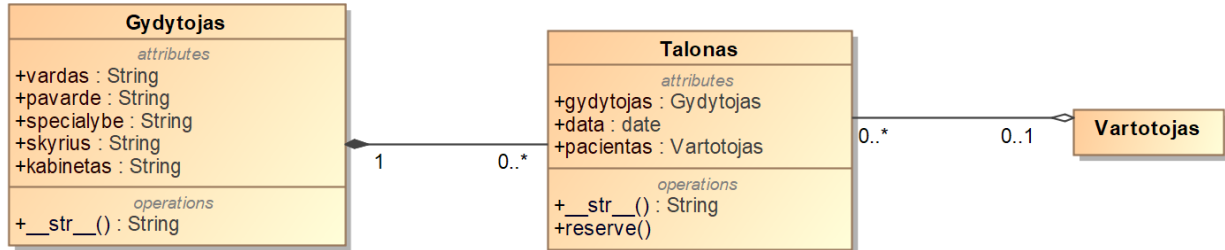
Su šiuo sistemos modeliu rašytojams iškilo daugiausiai problemų. Bibliotekos, naudotos kuriant prototipą, įgyvendino daugumą sudėtingesnių užduočių, kylančių kuriant ligoninės registracijos sistemą. Tarp bibliotekos įgyvendintų užduočių patenka:

- Vartotojo klasės įgyvendinimas
- Vartotojo registracija/prisijungimas
- Administratoriaus prisijungimas/sukūrimas
- Klasių objektų kūrimas/redagavimas/naikinimas

Dėl šios priežasties šie programos aspektai šiame modelyje nebus nagrinėjami.

2.1. Daktaro vizito registracija

4 diagrama. Daktaro vizito registracijos klasių diagrama



„Gydytojas“ klasė yra skirta laikyti informacijai, susijusiai su gydytoju. Gydytojas nėra nei sistemos vartotojas, nei sistemos prižiūrėtojas. Kuriant „Gydytojas“ klasę, privaloma nurodyti vardo bei pavardės laukų reikšmes.

`__str__()` metodai „Gydytojas“ bei „Talonas“ klasėse skirti atspausdinti klasės pavadinimui, remiantis nurodytais laukais rašant klases.

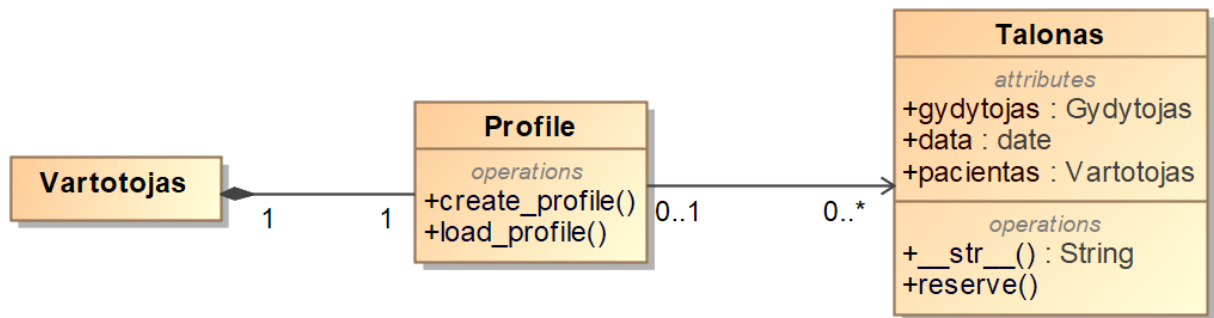
„Talonas“ klasė atitinka gydytojo registracijos taloną. Ji yra sąsaja tarp gydytojo ir vartotojo.

Metodas `reserve()` atlieka vartotojo priskyrimą prie gydytojo registracijos talono.

„Vartotojas“ klasė yra palikta tuščia, nes yra įgyvendinta bibliotekos. Ši klasė atitinka ir vartotoją, ir administratorių. Skirtumas tarp vartotojo ir administratoriaus programos lygmenyje toks, kad administratorius yra vartotojas, kuris turi ir sistemos prižiūrėtojo teises. Administratorius gali atliktai tai, ką gali atlikti ir vartotojas, bet vartotojas negali atlikti to, ką gali administratorius, pvz. sukurti naują gydytoją. Šio projekto atžvilgiu administratorius nesiekia to, ko siekia vartotojas.

2.2. Vartotojo rezervuotų daktaro talonų peržiūra

5 diagrama. Vartotojo rezervuotų daktaro talonų peržiūros klasių diagrama

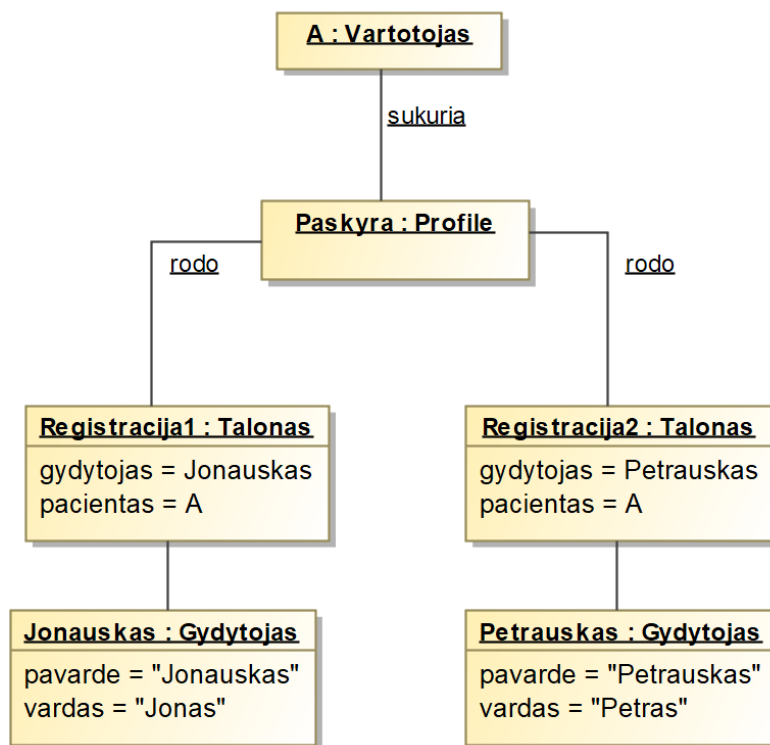


„Profile“ klasė skirta parodyti vartotojo rezervuotus buvusius bei esamus daktaro talonus. Savo viduje ji nelaiko „Talonas“ objektų.

Metodas `create_profile()` vykdomas, kai sukuriamas vartotojas. Šis metodas sukuria vartotojo profilį.

Metodas `load_profile()` vykdomas, kai vartotojas nori pažiūrėti savo profilį. Metodas suranda visus vartotojo rezervuotus gydytojo talonus bei juos atspausdina vartotojui.

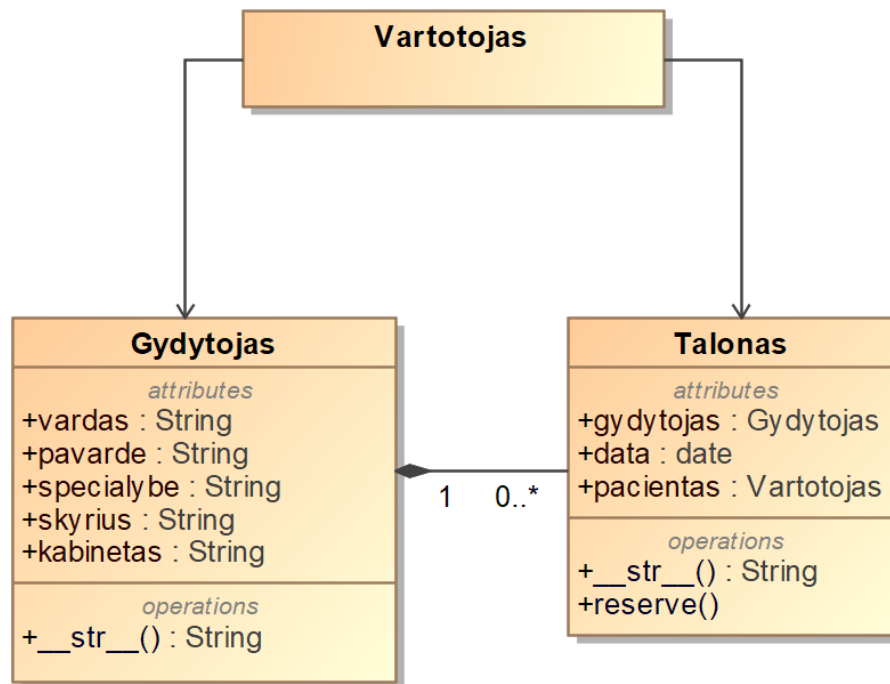
6 diagrama. Vartotojo rezervuotų daktaro talonų peržiūros objektų diagrama



Atlikus „Profile“ klasės `load_profile()` metodą, Vartotojas gauna visus talonus, kuriuose jis yra priskirtas kaip pacientas. Šių talonų objektas savyje nelaiko, o tik rodo gautą informaciją paieškos metu.

2.3. Daktaro informacijos peržiūra

7 diagrama. Daktaro informacijos peržiūros klasės diagrama

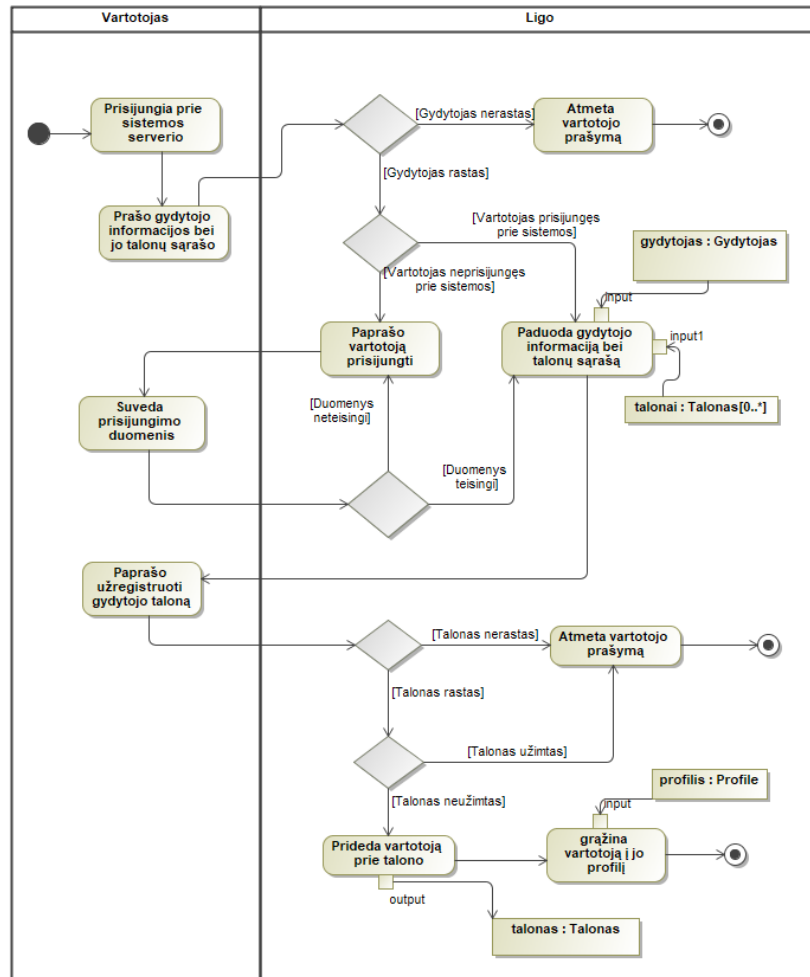


Sistemos naudotojas gali peržiūrėti daktaro informaciją, bet tik sistemos prižiūrėtojas gali keisti viduje esančią informaciją.

3. Dinaminis programų sistemos modelis

3.1. Daktaro vizito registracija

8 diagrama. Daktaro vizito registracijos veiklos diagrama



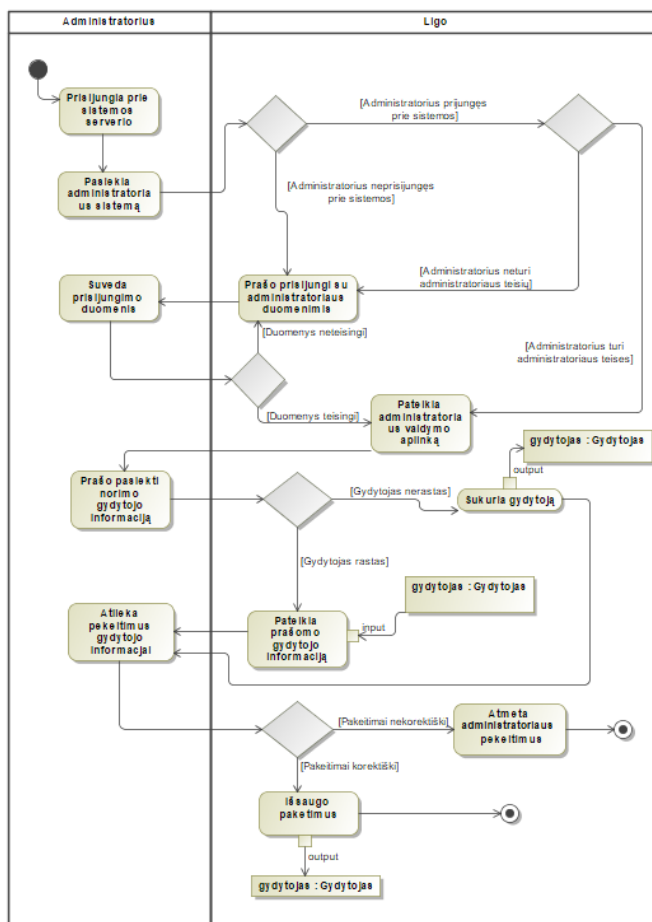
Diagramoje pavaizduotas prisijungimas prie sistemos serverio priylgsta pirminei prieigai prie sistemos (pvz. prisijungimui prie tinklapio su interneto naršykle). Prisijungimas prie sistemos atlieka vartotojo autentikavimo funkciją bei prileidžia jį prie jam leidžiamos pasiekti informacijos.

Šioje diagramoje pavaizduotas jungimosi procesas yra supaprastintas, nes nesigilina į vartotojo atpažinimą. Tai yra atliekama už projekto ribų esančių bibliotekų dėka, todėl neįtraukiama į diagramą.

Šioje diagramoje pavaizduotas ir daktaro duomenų peržiūra, todėl atskirai nebraižoma.

3.2. Informacinis sistemos prižiūrėjimas

9 diagrama. Gydytojo informacijos kūrimo bei redagavimo veiklos diagrama

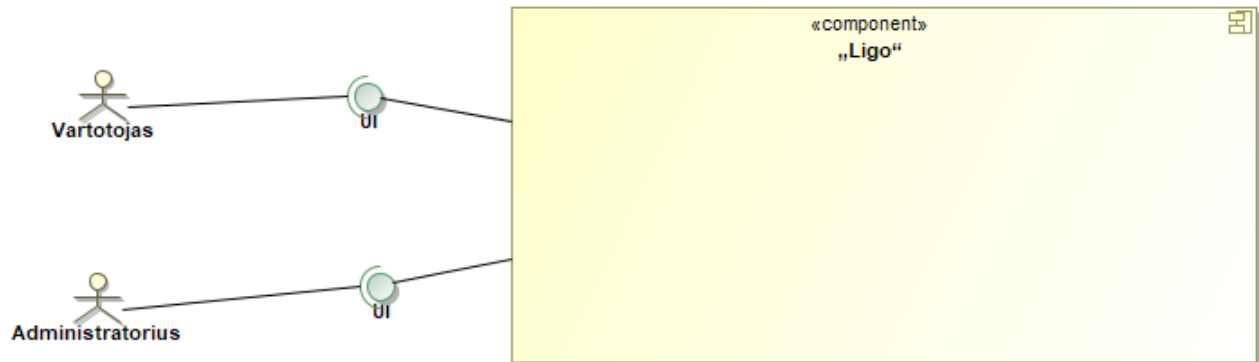


Administratoriaus teises šioje diagramoje atitinka sistemos prižiūrėtojo pareigas. Jei prie administratoriaus sistemos bando prisijungti sistemos naudotojas, sistema atmeta jo prisijungimą ir prašo administratoriaus duomenų, tiksliau duomenų, esančių „Vartotojas“ objekte, kuris atlieka sistemos prižiūrėtojo vaidmenį.

Ši veiklos diagrama atitinka ir gydytojų talonų kūrimą bei redagavimą. Esminis skirtumas, lyginant su šia diagrama yra tai, kad vietoje gydytojo informacijos bei „Gydytojas“ klasės yra talono informacija bei „Talonas“ klasė. Dėl šios priežasties autoriai pasirinko nevaizduoti šios diagramos

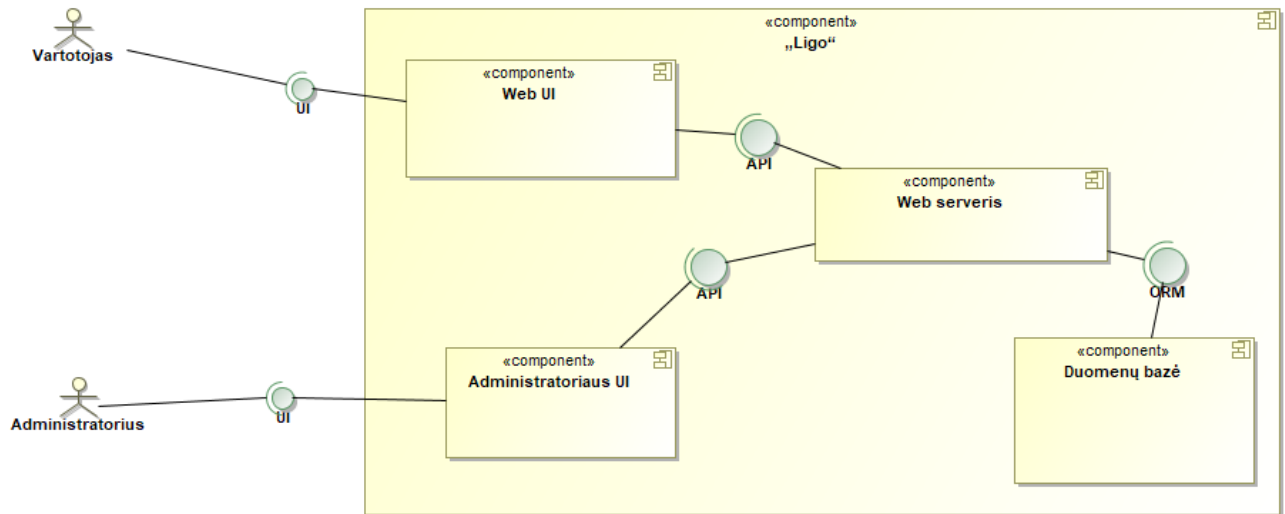
4. Programų sistemos komponentai

10. diagrama. Pavaizduota **sistemos sąveika su vartotoju**



Sistema „Ligo“ su vartotoju bei administratoriumi bendrauja per internetinį tinklapį. Vartotojas pasiekia sistemą per pagrindinį tinklapio puslapį, kol administratorius naudoja specialų administratoriaus valdymo aplinką.

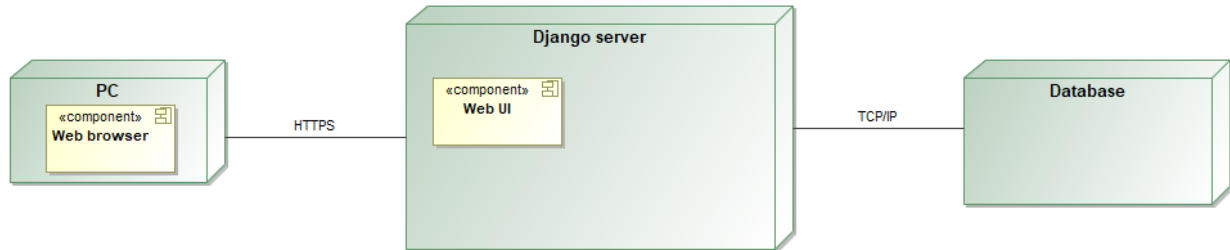
11 diagrama. Pavaizduota serverio sąveika su duomenų baze ir vartotoju



Web serveris pasiekiamas prisijungus prie vietinio tinklo. Serveris bendrauja su duomenų baze ir ten saugo vartotojų ir darbuotojų informacija. Sėkmingai prisijungus administratoriui, jam rodomas išorinis interfeisas. Prisiregistravęs vartotojas turi galimybę užsiregistruoti gydytojo vizitui gaudamas taloną. Vartotojas užregistruojamas pas gydytoją ir informacija užsaugoma duomenų bazėje.

5. Komponentų išsidėstymas tinkle

. 12 diagrama. diagrama, rodanti fizini ryšį tarp serverio ir vartotojo



Diagramoje atvaizduotas fizinis pjūvis. Vartotojas naudodamasis lokaliu prietaisu gali pasiekti Django serverį. Django serveriai įgyvendina tinkle apsaugą dėl to duomenys perduodami HTTPS protokolu. Prie nuotolinio serverio gali prisijungti daug lokalių prietaisų. Vartotojas interneto naršykle prisijungęs prie tinklapio gali matyti visą programos teikiama funkcionalumą. Jei klientas nori išsaugoti naują informaciją, tai tinklapio naudotojo interfeiso komponento pagalba jis naviguoja į reikiamą langą ir jame atlieka išsaugojimo veiksmus, kurie yra siunčiami ir užregistruojami, duomenys yra išsaugomi į duomenų bazę. Serveris susisieikia su duomenų baze ir keičiasi informacija. Django serveris laiko informaciją apie kiekvieną vartotoją ir ją saugo

Terminų žodynas

- Sistemos naudotojas – Aktorius, turintis teisę redaguoti, kurti, peržiūrėti tik jam skirtą informaciją arba bendrai prieinamą informaciją.
- Sistemos prižiūrėtojas – Aktorius, turintis teisę redaguoti, kurti, peržiūrėti visą sistemos turimą naudotojų informaciją.
- Gydytojo registracijos talonas – Vizito pas daktarą nurodytu laiku nurodytam naudotojui objektas (šioje sistemoje atitinka „Talonas“ modelį).

Literatūros sąrašas

Phillipe Kruchten. The „4+1“ View Model of Software Architecture.
<https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>

Priedai

1. Saugumo reikalavimai

Ligoninės sistema duomenų bazėje saugo vartotojų ir dirbančių darbuotojų asmeninius duomenis, todėl šios informacijos negali pasiekti paprasti vartotojai. Taip pat, registruojantis, sistema reikalauja tenkinti slaptažodžio stiprumo kriterijus, kas apsaugo nuo įsibrovėlių. Vedant slaptažodį, jis nėra matomas.

Sistema yra apsaugota nuo įvairių pavojų :duomenų bazės įsiveržimo, Cross-site scripting (XSS), Cross-site request forgery (CSRF), Clickjacking protection. Be to puslapio valdytojo galimybės ir informacija yra nepasiekiamos kitiems žmonėms