

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

## **Stalo žaidimų programėlė**

### **Board Games Application**

Programų sistemų inžinerija: 1 laboratorinis darbas

Atliko:	2 kurso 5 grupės studentai	
	Elena Reivytytė	(parašas)
	Matas Šilinskas	(parašas)
	Kasparas Taminskas	(parašas)
	Aidas Vaikšnoras	(parašas)
	Tadas Žaliauskas	(parašas)
Darbo vadovas:	dr. Vytautas Valaitis	(parašas)

## **Anotacija**

Šiame dokumente pristatomas mobiliosios stalo žaidimų programėlės Board Games 4+1 architektūrinis modelis, kurį sudaro loginis, kūrimo, fizinis, procesų ir užduočių pjūviai, visi kartu parodantys sistemą iš skirtingų pusių, papildantys vieni kitus. Modelio pagrindas - standartinės UML diagramos.

## TURINYS

ANOTACIJA .....	1
ĮVADAS .....	3
1. LOGINIS PJŪVIS .....	4
1.1. Klasų Diagramos .....	4
1.2. Objektų diagrama .....	7
2. UŽDUOTYS .....	8
2.1. Svečio ir registruoto vartotojo užduotys .....	8
2.2. Žaidimo dalyvio ir šeimininko užduotys .....	9
2.3. Sistemos administravimas .....	10
3. KŪRIMO PJŪVIS .....	11
3.1. Bendroji UML diagrama.....	11
3.2. Vartotojo komponentų diagrama .....	11
3.3. Detalesnė vartotojo komponentų diagrama .....	12
4. FIZINIS PJŪVIS .....	14
4.1. Fiziniai mazgai tinkle, jų komunikacija .....	14
4.2. Artifaktų diegimas į mazgus .....	15
4.3. Topologinio išdėstymo tinkle pavyzdys .....	16
5. PROCESŲ PJŪVIS .....	17
5.1. Prisijungimas .....	17
5.2. Pagrindinis navigacijos langas.....	19
5.3. Žaidimo sukūrimo langas .....	20
5.4. Prisijungimo prie egzistuojančio žaidimo langas .....	21
5.5. Draugų sąrašo langas.....	22
5.6. Žaidimo langas .....	23
LITERATŪROS SĄRAŠAS .....	24

# **Įvadas**

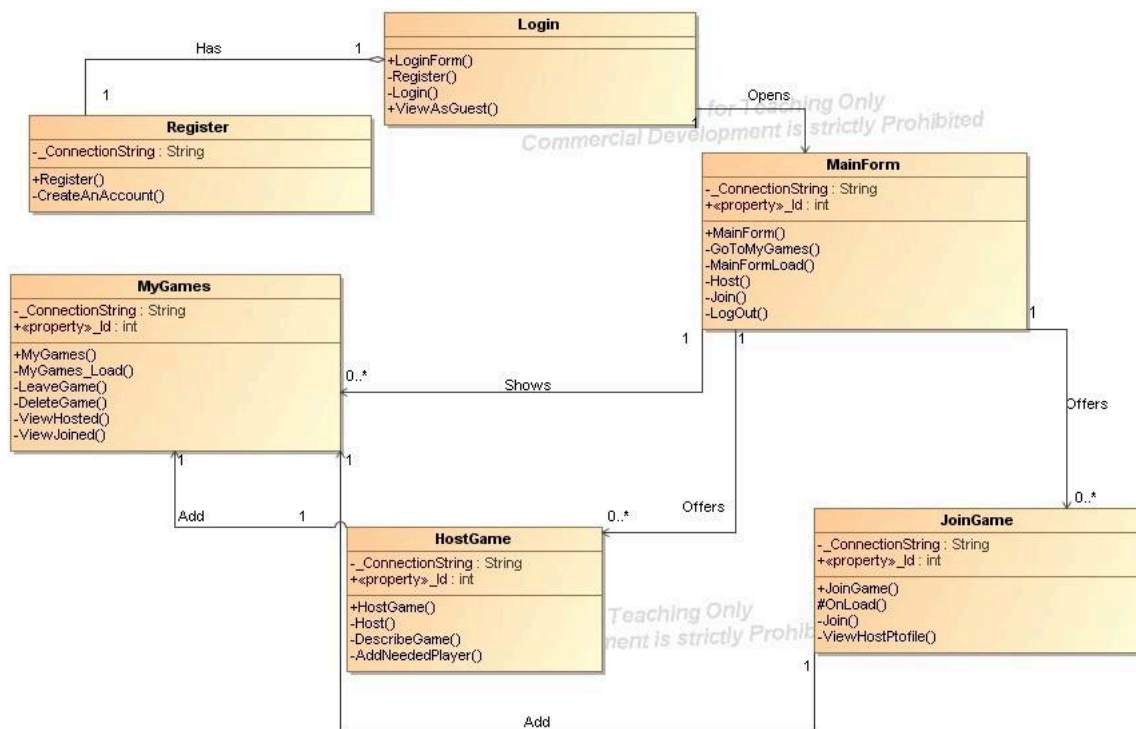
Board Games - tai aplikacija, sujungianti norinčius žaisti stalo žaidimus žmones su tais, kuriems trūksta žaidėjų.

# 1. Loginis pjūvis

## 1.1. Klasių Diagramos

Tam, kad būtų patogiau suprasti, klasių diagrama yra padalinta į kelias dalis:

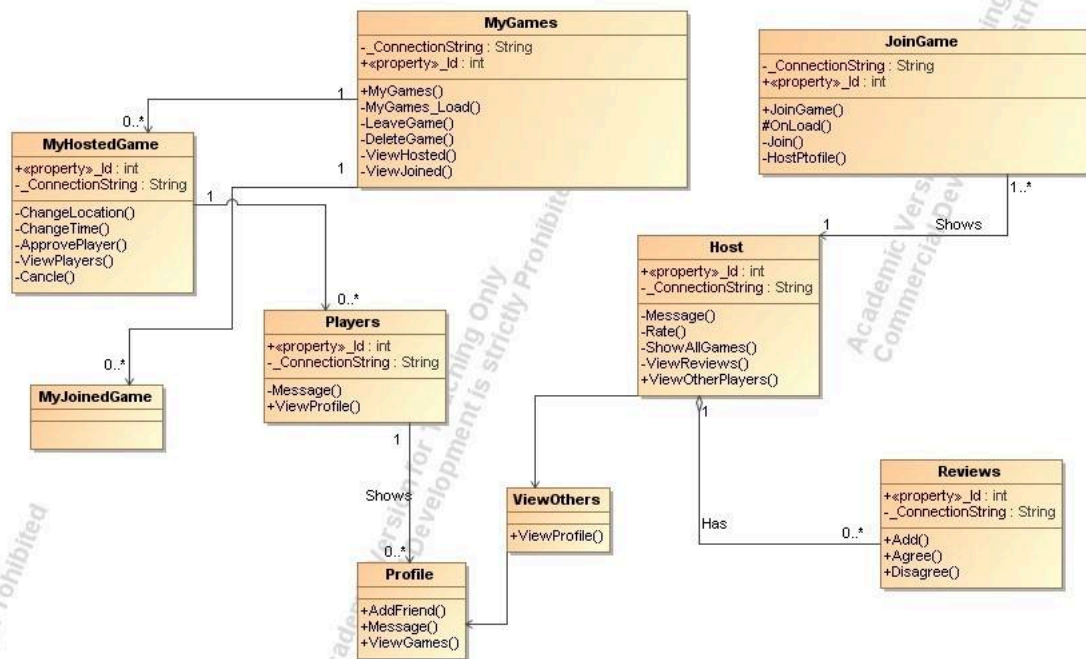
Pirmoji klasių diagramos dalis :



1 pav. Klasių diagrama: registracija ir pgr funkcijos

Klasė **Login**, atsakinga už vartotojo autentifikaciją, turi registracijos langą (klasę **Register**). Viena **Login** klasė gali turėti vieną registracijos klasę. Užsiregistravus, **Login** klasė gali atidaryti pagrindinį vartotojo puslapį (**MainForm** klasę), kuris gali parodyti **MyGames** klasę, t.y. vartotojo žaidimus, siūlo sukurti žaidimą (**HostGame**) arba jungtis prie jau egzistuojančio (**JoinGame**) žaidimo. Kiekviena **MainForm** klasė gali parodyti, sukurti ir prisijungti prie kiek norimai daug žaidimų. Kiekvieną kartą sukūrus ar prisijungus prie žaidimo, jis yra pridedamas prie **MyGames**.

## Antroji klasių diagramos dalis :



2 pav. Klasių diagrama: žaidimų kūrimas ir prisijungimas

Klasė MyGames atveria vartotojo sukurtus žaidimus (klasė MyHostedGame) bei žaidimus, prie kurių jis prisijungė (MyJoinedGame). Abiejų variantų vartotojas gali turėti kiek norimai daug. Kiekvienas sukurtas žaidimas gali atverti prisijungusių kitų žaidėjų sąrašą ir iš jo yra pasiekiamas kiekvieno žaidėjo profilis (klasė Profile). Jei vartotojas pasirinko prisijungti prie žaidimo, už tai yra atsakinga JoinGame klasė, kurios vieną ar daugiau elementų (žaidimų) sukūrė konkretus kūrėjas (klasė Host). Host turi klasę Reviews, nes kiekvienas kūrėjas gali turėti atsiliepimų. Prisijungus prie žaidimo, klasė ViewOthers užtikrina galimybę peržvelgti kitų prisijungusių žaidėjų sąrašą ir jų profilius (Profile).

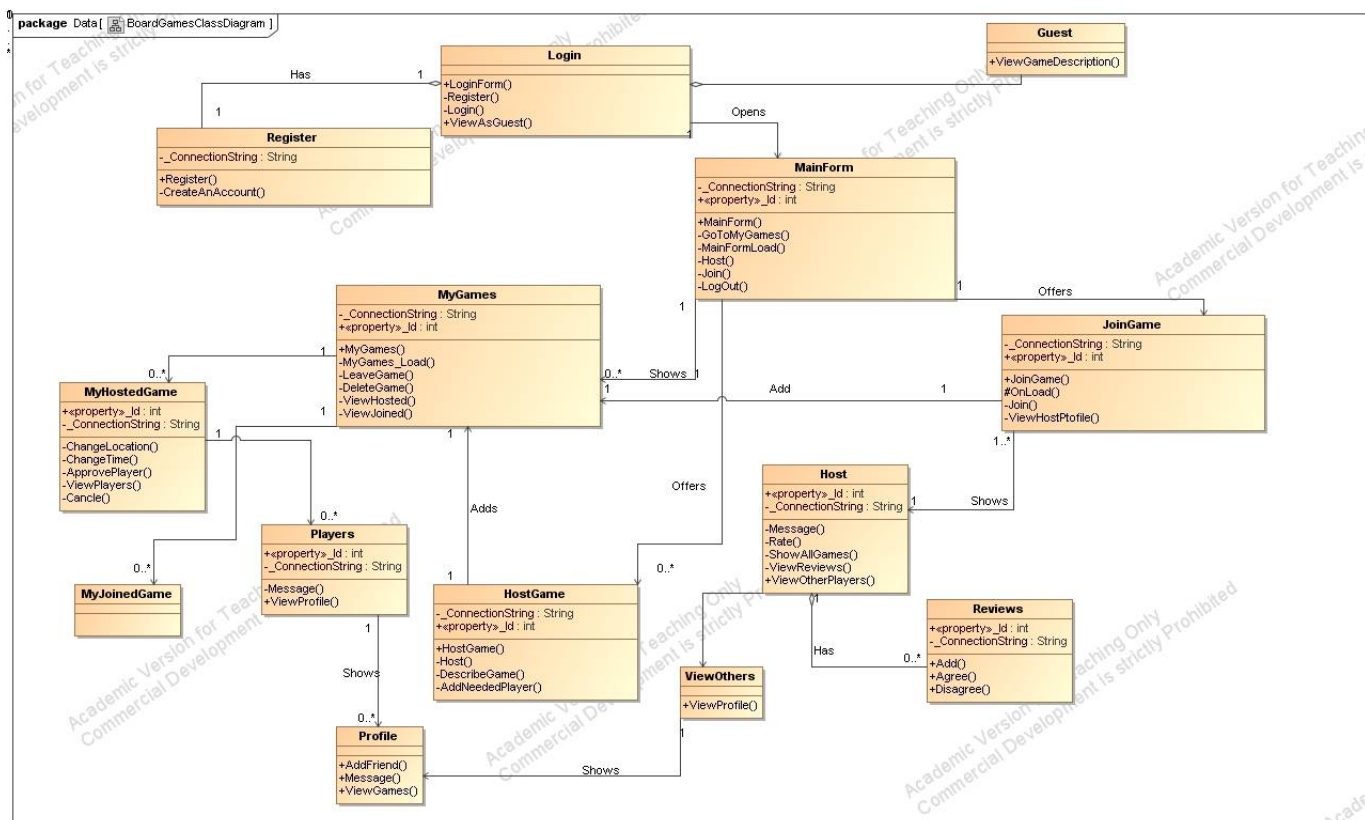
## Trečioji klasių diagramos dalis :



3 pav. Klasių diagrama: Svečias

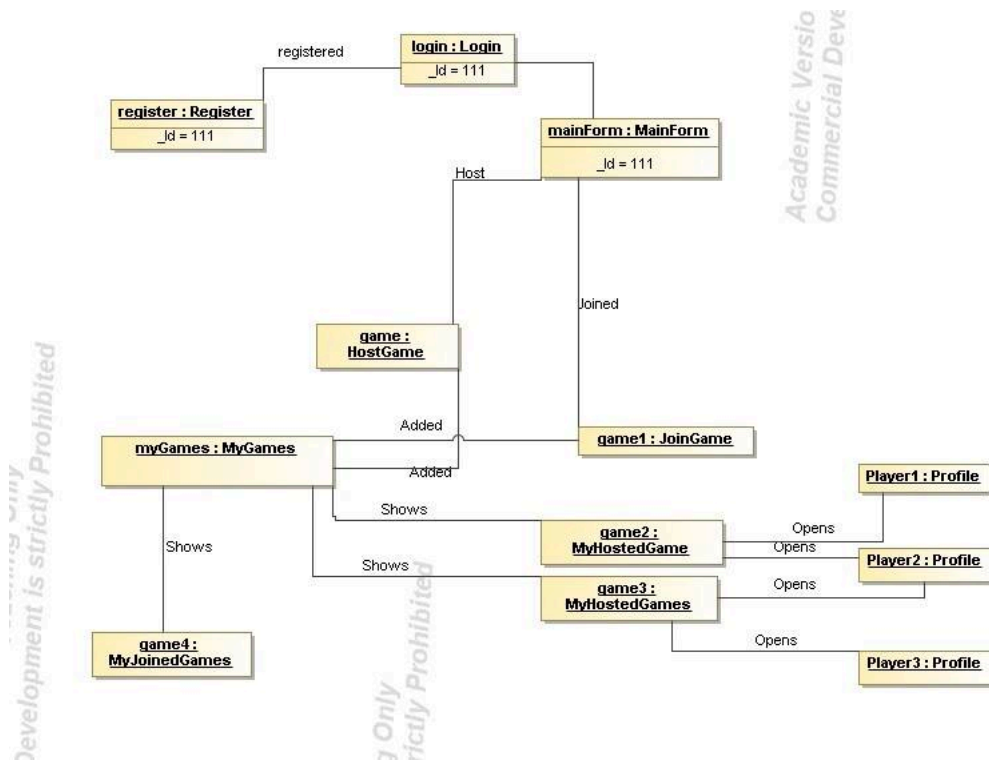
Jei vartotojas nėra prisiregistravęs (Guest), Login forma gali nuvesti jį prie sukurtų žaidimų sąrašo.

## Pilna klasių diagrama :



4 pav. Visa klasių diagrama

## 1.2. Objektų diagrama



5 pav. Objektų diagrama

Užsiregistravęs vartotojas yra autentifikuojamas ir nusiunčiamas į savo pagrindinį puslapį mainForm. Jis buvo sukūręs žaidimą game ir prisijungęs prie žaidimo game1, kurie abu buvo pridėti prie jo asmeninių žaidimų sarašo myGames. Jame jau yra žaidimai game2 ir game3 (kuriuos jis sukūrė pats) bei game4, prie kurio buvo prisijungęs. Kitas vartotojas player1 buvo prisijungęs prie žaidimo game2, player2 - prie game2 ir game3, o player3 - prie game3.



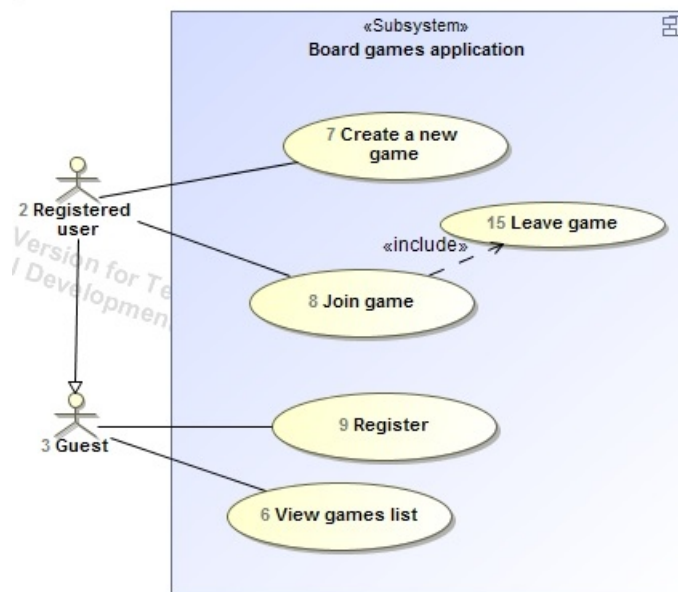
## 2. Užduotys

Pagrindinės sistemos užduotys ir su jomis susiję agentai

- Agentai:
  - **Svečias.** Neprisijungęs programėlės naudotojas.
  - **Registruotas vartotojas.** Žmogus, sėkmingai atlikęs registraciją ir prisijungęs prie programėlės, galintis dalyvauti jos veikloje.
  - **Žaidimo dalyvis.** Registruotas vartotojas, nusprendęs prisijungti prie vieno iš sistemos siūlomų žaidimų.
  - **Žaidimo šeimininkas.** Registruotas vartotojas, pridėjęs žaidimą sistemoje.
  - **Administratorius.** Žmogus, atsakingas už tinkamą sistemos darbo palaikymą.

Pagrindinės užduotys

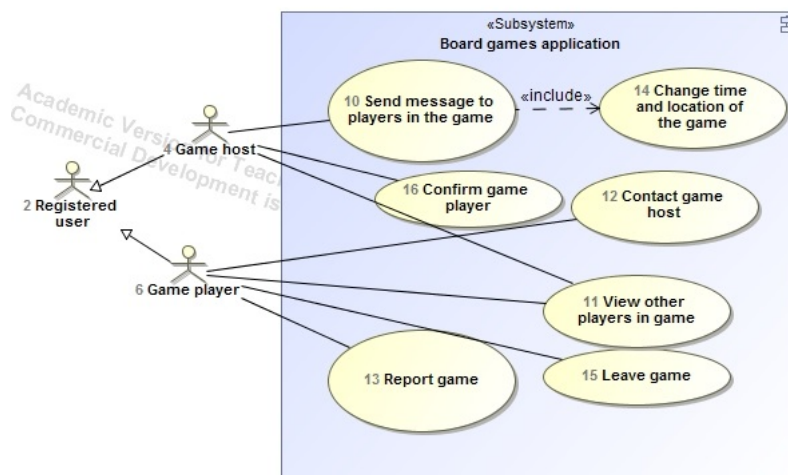
### 2.1. Svečio ir registruoto vartotojo užduotys



6 pav. Svečio ir registruoto vartotojo užduotys

Kol vartotojas nėra užsiregistravęs sistemoje, jam suteikiamos svečio teisės, kurios leidžia tik peržiūrėti sukurtus žaidimus. Norėdamas įgyti daugiau privilegijų, vartotojas privalo užsiregistruoti. Registruotas vartotojas jau gali dalyvauti programėlės veikloje: sukurti naują žaidimą ir apie jį paskelbti, arba prisijungti prie jau esamo.

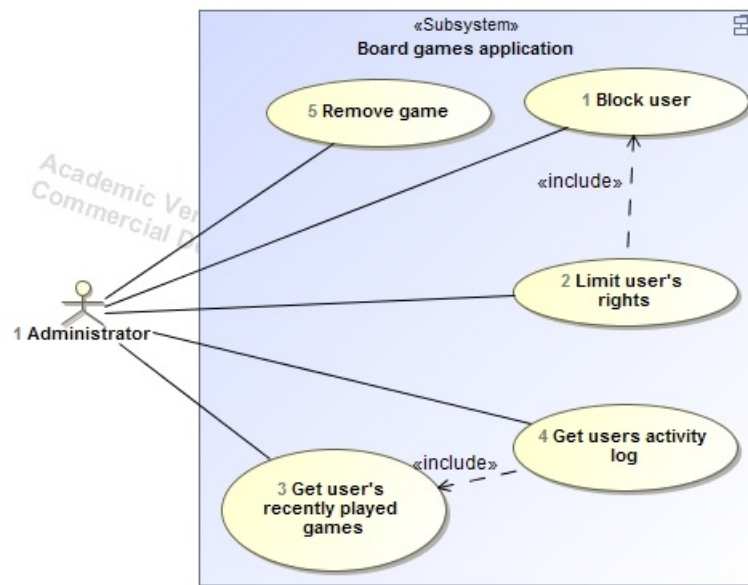
## 2.2. Žaidimo dalyvio ir šeimininko užduotys



7 pav. Žaidimo dalyvio ir šeimininko užduotys

Registruotas vartotojas gali atlikti du vaidmenis: arba būti sukurtu žaidimo šeimininku, arba paprasčiausiu dalyviu. Žaidimo dalyviui leidžiama pranešti sistemos administratoriui apie nevystantį žaidimą, tuo atveju, jeigu žaidėjas nuvyko į sutartą žaidimo vietą reikiamu laiku ir jam nepavyko surasti kitų užsiregistravusių dalyvių. Jeigu bent vienas žaidėjas (išskyrus šeimininką) nesutinka su tokia informacija, pranešimas laikomas melagingu. Priešingu atveju, sulaukus administratoriaus pritarimo, žaidimas išimamas iš aktyvių žaidimų sąrašo norint užkirsti tolimesnę galimą dalyvių pritraukimą. Tai pat, šeimininkui patvirtinus žaidėją, jis įgauna teisę peržiūrėti kitų, jau pareiškusių norą dalyvauti žaidėjų profilius ir su jais susipažinti naudojantis programėlės asmeninių žinučių sistema. Norėdamas pasiteirauti dėl žaidimo detalių arba visais kitais iškilusiais klausimais, jis taip pat gali tiesiogiai kreiptis į žaidimo šeimininką. Šeimininkas gali informuoti žaidėjus apie nenumatytai pasikeitusią žaidimo vietą ar laiką, susisiekti su jais ir iš anksto trumpai supažindinti su žaidimo taisyklėmis. Jeigu žaidimo dalyviui nepatinka kitų dalyvių kompanija ar bet kokios kitos žaidimo detalės, jis gali bet kada palikti žaidimą, o žaidimo šeimininkas apie tai yra informuojamas.

## 2.3. Sistemos administravimas



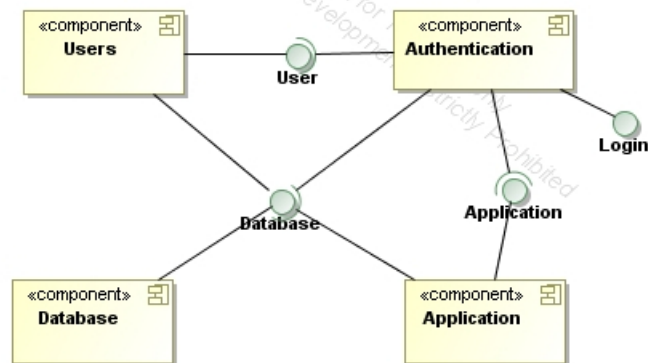
8 pav. Sistemos administravimas

Administratorius prižiūri programėlės pateikiamos informacijos patikimumą. Jis turi teisę ištrinti registruoto vartotojo sukurtą žaidimą siekdamas apsaugoti dar vėliau prie žaidimo galimai prisijungusiančius dalyvius tuo atveju, jei vartotojo pateikti duomenys nėra tikslūs ar apie žaidimą jau buvo gautas įspėjamas pranešimas iš kitų sąžiningų vartotojų. Jeigu vartotojas nesiliauja kurti fiktyvių žaidimų, jam gali būti taikomos nuobaudos: laikinai (pvz. parai) apribojimas organizuoti bet kokius žaidimus. Jei tai kartojasi, administratorius gali apsvarstyti pasiūlymą užblokuoti vartotojo paskyrą ir įtraukti jį į nepageidaujamų sąrašą. Kaip pagalbinę priemonę vartotojų veiksmams sekti administratorius gali naudoti vartotojų veiklos žurnalą, kuriame būtų pateikiami chronologiškai surikiuoti vartotojų veiksmai, gauti nurodžius dominantį periodą. Veiklos žurnalą būtų galima filtruoti pagal konkretų vartotoją, taip susiaurinant paiešką.

### 3. Kūrimo pjūvis

Komponentų dekompozicija įgyvendinta taip: nuo bendresnių komponentų pereinama iki detalesnių.

#### 3.1. Bendroji UML diagrama

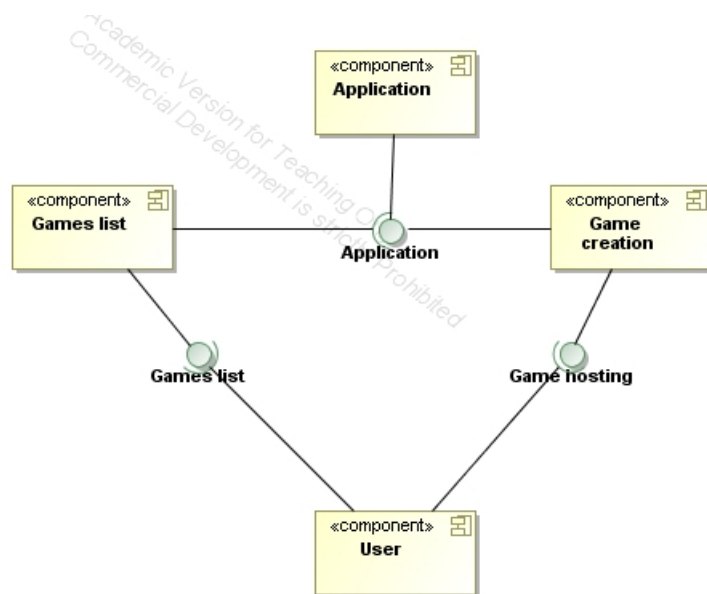


9 pav. Bendroji UML diagrama

Vartotojui įsijungus stalo žaidimų aplikaciją, jis prie jos turi prisijungti, kad būtų patvirtinta jo tapatybė. Tai atliekama prisijungimo sąsajoj, kurią suteikia autentifikacijos komponentas. Patikrinus duomenis duomenų sistemoje, kurioje saugoma informacija apie vartotoją bei jo įvertinimą, gautą iš kitų žaidėjų, ir sėkmingai autentifikavus vartotoją, jam leidžiama naudotis stalo žaidimų aplikacijos ir naudotojo komponentų teikiamomis funkcijomis. Naudotojo komponento teikiama sąsaja leidžia pateikti informaciją apie save kitiems, o aplikacijos komponento sąsaja suteikia galimybę kurti žaidimus, prisijungti prie jų, vertinti vartotojus, pateikti informaciją apie esančius žaidimus pačiam vartotojui ar pačio kuriamus kitiems vartotojams.

#### 3.2. Vartotojo komponentų diagrama

Aukščiau pateikta UML diagrama, kurioje vaizduojami bendri komponentai. Atskirus sistemos komponentus patogiau vaizduoti atskirai, todėl tam sukurtos papildomos diagramos.

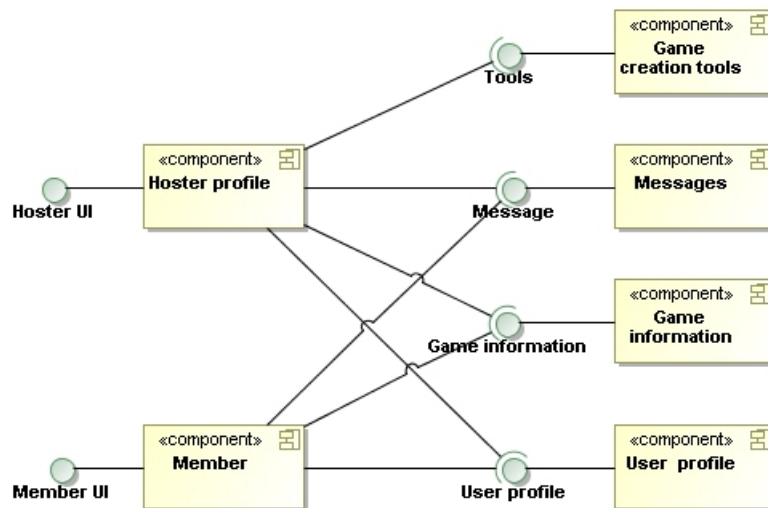


10 pav. Vartotojo komponentų diagrama

Vartotojui pateikiama atitinkama sąsaja priklausomai nuo to, ar vartotojas pasirinko būti žaidimo kūrėju, ar prisijungti prie žaidimo. Jei vartotojas pasirinko prisijungti prie esamo žaidimo, tai jam yra pateikiama žaidimų sąrašo sąsaja, pateikianti sąrašą žaidimų, kuriuose trūksta narių. Jei vartotojas nusprendė pats kurti žaidimą, tada jis nukeliamas į žaidimo kūrimo langą ir įgauna papildomų funkcijų. Tai suteikia žaidimo kūrimo komponentas ir jo suteikiama žaidimo organizavimo sąsaja. Pasirinkimą būti žaidimo kūrėju ar paprastu nariu bet kada galima pakeisti.

### 3.3. Detalesnė vartotojo komponentų diagrama

Kadangi vartotojai gali arba prisijungti prie žaidimų, arba būti jų kūrėjais, jiems reikalingi atitinkami komponentai. Kad geriau matytųsi, kokie iš jų priklauso kiekvienam tipui, sukurta atskira UML diagrama.



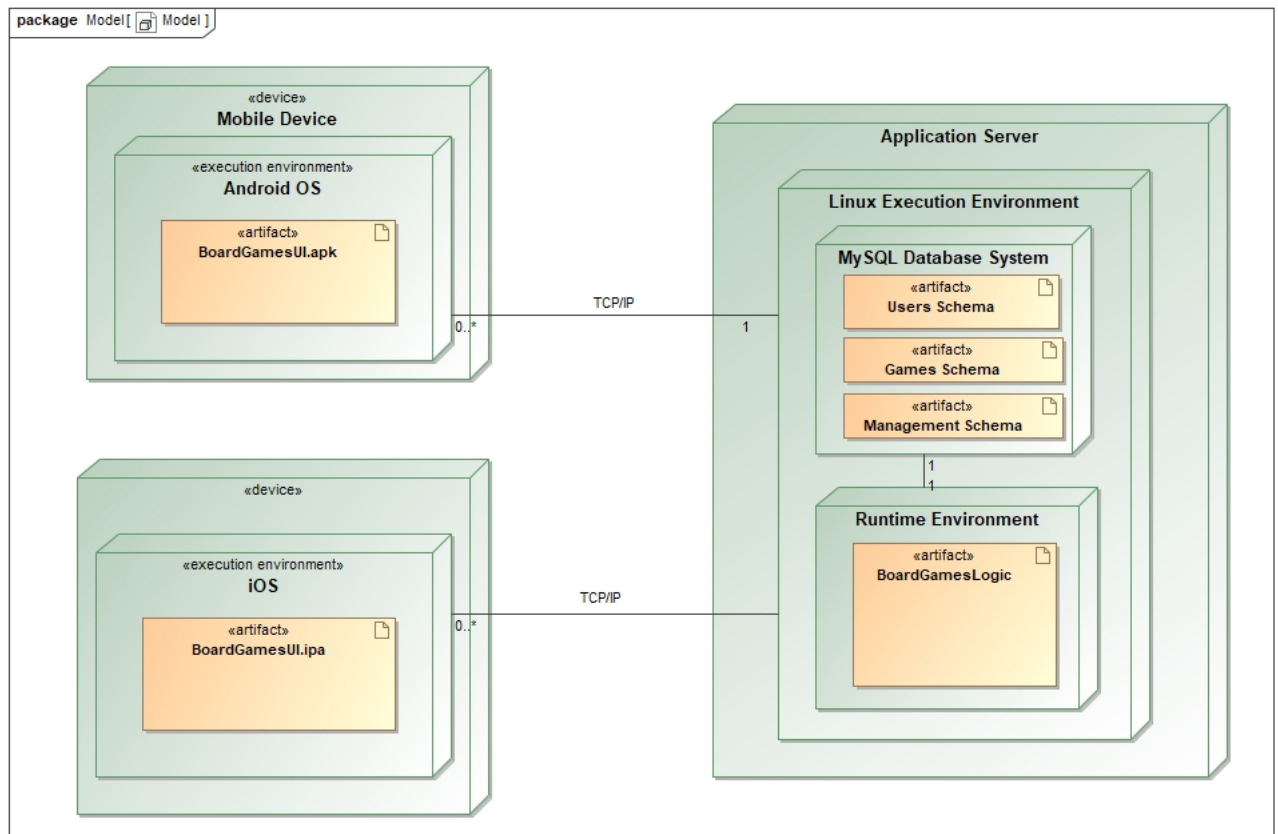
11 pav. Detalesnė vartotojo komponentų diagrama

Tiek paprasti nariai, tiek kūrėjai gali naudotis vartotojo profilio, žaidimo informacijos ir žinučių komponentais, kurie suteikia atitinkamas sąsajas. Vartotojo profilio komponentas leidžia parodyti informaciją apie žaidimuose esančius narius bei jų įvertinimą ir patikimumą. Žaidimo informacijos komponente trumpai nupasakojama apie žaidimą bei pateikiamas jo pavadinimas. Žinučių komponentas suteikia galimybę susisiekti su kiekvienu žaidimo nariu atskirai ir bendrauti bendrame grupės pokalbyje. Kūrėjas taip pat gali naudotis žaidimo kūrimo įrankių komponentu. Jam suteikiamos galimybės pateikti ir keisti žaidimo informaciją, priimti, pakviesti bei išmesti žaidimo narius.

## 4. Fizinis pjūvis

Šioje dalyje pateikta informacija apie sistemos fizinius mazgus, juose įdiegtas vykdomasis aplinkas, reikalingas vykdyti programos komponentų darbą, taip pat apie komunikaciją tarp mazgų ir juose įdiegtus artifaktus, kurie saugo programinį kodą.

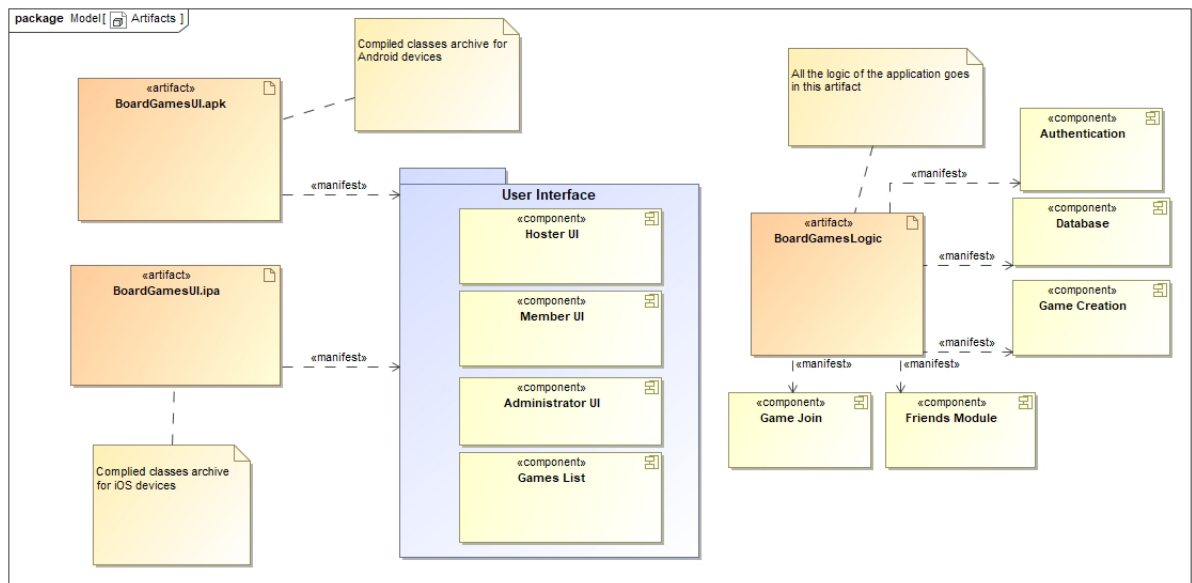
### 4.1. Fiziniai mazgai tinkle, jų komunikacija



12 pav. Fizinių sistemos mazgų išdėstymo tinkle diagrama

Mobilioji aplikacija dėl savo paprasto loginio pobūdžio nereikalauja kompleksiškos tinklo architektūros su daugybe fizinių komponentų. Išdėstymo diagramoje pateiktas kliento mazgas – mobilusis įrenginys – kuriame veikia Android OS arba iOS. Šiose operacinių sistemų aplinkose diegiamas mobiliosios aplikacijos artifaktas (.apk – Android atveju, .ipa – iOS atveju), kurį sudaro visos sukompiliuotos projekto vartotojo sąsajos klasės, resursai, taip pat ir nesukompiliuoti resursai. Kitas mazgas – nutolęs aplikacijos serveris, su kuriuo TCP/IP protokolu komunikuoja kliento mobilusis įrenginys. Šiame nutolusiame fiziniame serveryje veikia stabili Linux OS aplinkos distribucija, užtikrinanti duombazės pasiekiamumą ir darbo nepertraukiamumą. Šioje OS instaliuota MySQL duomenų bazių valdymo sistema, kurioje sukurtos trys lentelių schemas: Users (informacija, susijusi su aplikacijos naudotojais), Games (informacija, susijusi su žaidimais, jų sesijomis), Management (aplikacijos valdymo, statistinių duomenų schema). Taip pat šioje Linux aplinkoje veikia programėlės loginio kodo vykdomoji aplinka, kurioje įdiegtas loginis artifaktas, atsakingas už visą programėlės logiką.

## 4.2. Artifaktų diegimas į mazgus



13 pav. Artifaktų sandara ir diegimas į mazgus

Sistemą sudaro du pagrindiniai artifaktai - įdiegtas vartotojo sąsajos paketas, išdėstomas kliento mobiliajame įrenginyje, ir logikos paketas, įdiegtas nutolusio aplikacijos serverio kodo vykdomojoje aplinkoje.

### Mobiliajame įrenginyje įdiegto artifakto turinys:

- Žaidimo šeimininko vartotojo sąsajos komponentas
- Paprasto programėlės naudotojo vartotojo sąsajos komponentas.
- Administratoriaus vartotojo sąsajos komponentas
- Esamų žaidimų sąrašas ir informacija apie juos

### Nutolusiame serveryje įdiegto artifakto turinys:

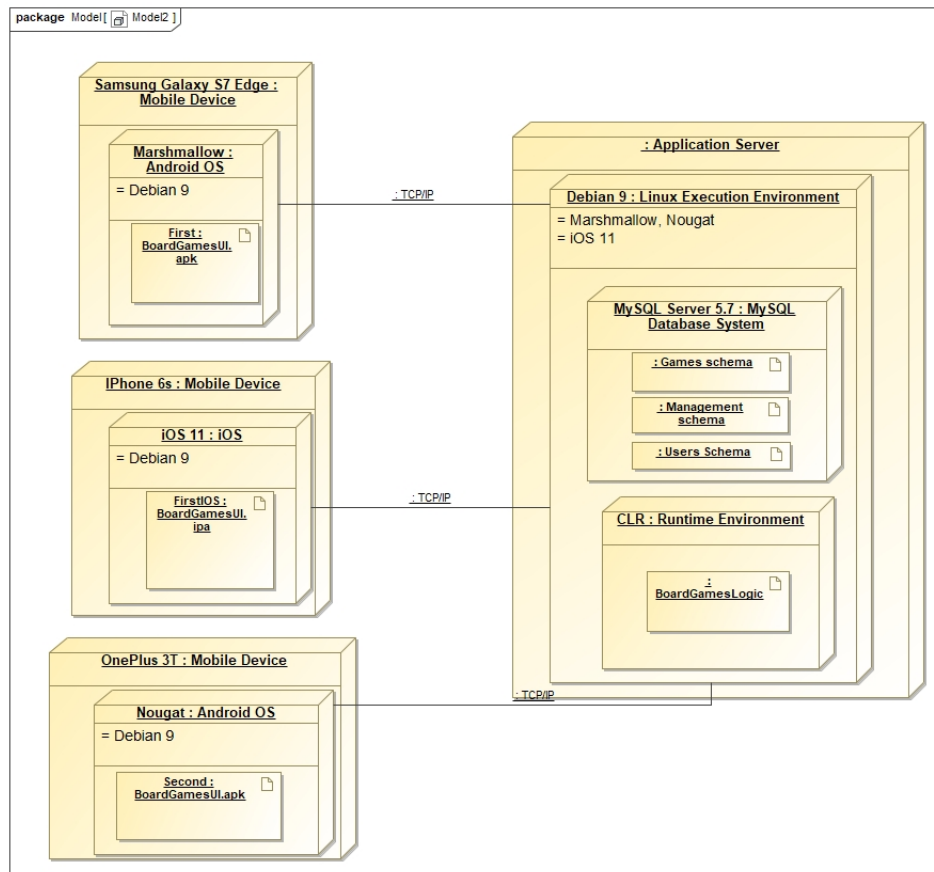
- Autentifikacijos komponentas - visa logika, susijusi su prisijungimu, registracija
- Duombazės komponentas - logika, kurioje apibrėžta komunikacija su DBVS
- Žaidimo sukūrimo logika - validacija, patvirtinimas, sistemos informavimas
- Prisijungimo prie žaidimo logika - validacija, žaidimo šeimininko, kitų žaidėjų informavimas
- Draugų modulio komponentas - visa logika, apibrėžianti galimą tarpusavio komunikaciją tarp programėlės naudotojų

Taip pat nutolusiame serveryje veikiančios Duomenų bazės valdymo sistemos aplinkoje įdiegti šie artifaktai, apibrėžiantys loginį duomenų bazės lentelių išskaidymą:



- Naudotojų lentelių schema
- Žaidimų lentelių schema
- Valdymo, priežiūros lentelių schema

#### 4.3. Topologinio išdėstymo tinkle pavyzdys



14 pav. Fizinio išdėstymo tinkle pavyzdys, komunikacija

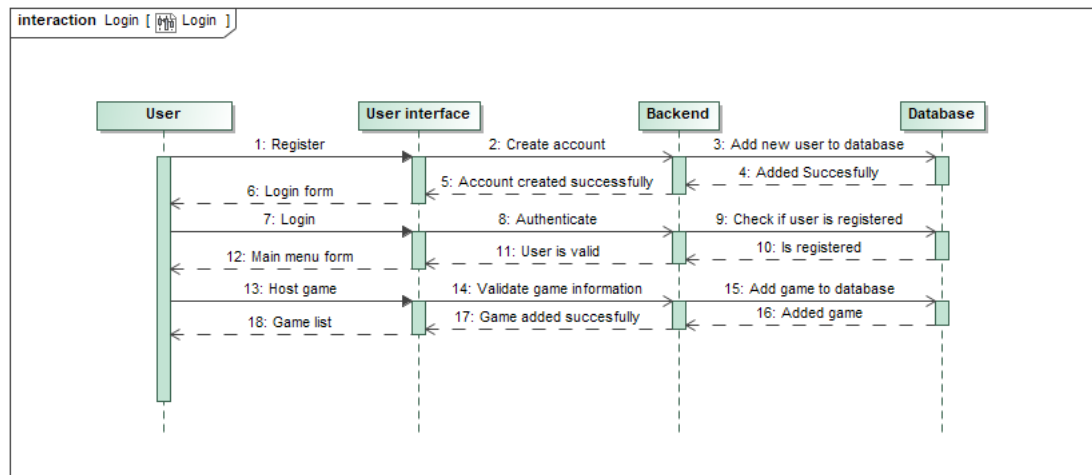
Diagramoje pavaizduotas galimas realaus sistemos vaizdo atvejis, kuris atitinka anksčiau aprašytą mazgų komunikacijos ir artifaktų diegimo logiką. Pavyzdyje pateikti trys klientiniai įrenginiai (2 Android, 1 iOS), kurie TCP/IP protokolo pagalba komunikuoja su konkrečiu Linux Debian 9.0 distribucijos serveriu, kuriame įdiegtas MySQL 5.7 versijos serveris, apdorojantis programėlės SQL užklausas ir Common Language Runtime vykdomoji aplinka, apdorojanti loginę programinę kodą, jį paverčianti mašininu, kurį vėliau įvykdo fizinis mazgas.

## 5. Procesų pjūvis

Aplikacijoje yra vartotojo sąsajos, duomenų apdorojimo ir duomenų bazės procesai. Vartotojas tiesiogiai bendraudamas su vartotojo sąsaja, siunčia asinchronines žinutes duomenų apdorojimo procesui, kuris savo ruožtu bendrauja su duomenų baze.

### 5.1. Prisijungimas

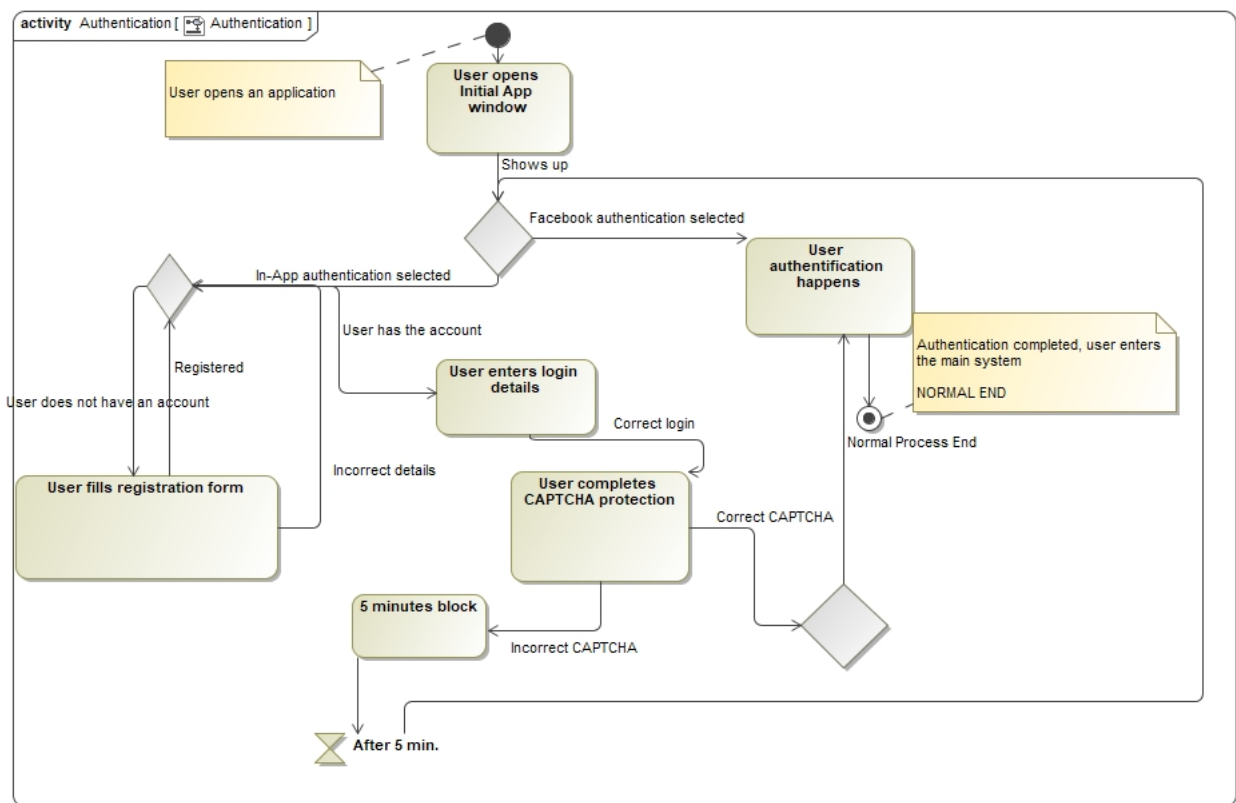
Prisijungimo procesų sekos diagrama :



15 pav. Prisijungimo procesų sekų diagrama

Visų pirma vartotojas privalo užsiregistruoti. Vartotojo sąsaja nusiunčia JSON žinutę su gautais duomenimis loginiam procesui, kuris patikrina ar duomenys yra tinkami ir prideda vartotoją į duomenų bazę. Užsiregistravęs vartotojas turi prisijungti, procedūra ganėtinai panaši - vartotojo sąsajoje gauti duomenys patikrinami ir jei jie yra validūs, patikrinama ar slaptažodis sutampa su esančiu duomenų bazėje. Autentifikuotas vartotojas yra prijungiamas prie programėlės. Jei slaptažodis netinka arba vartotojas nerastas duomenų bazėje, leidžiama patikslinti prisijungimo laukus ir prisijungimo procesas kartojamas dar kartą.

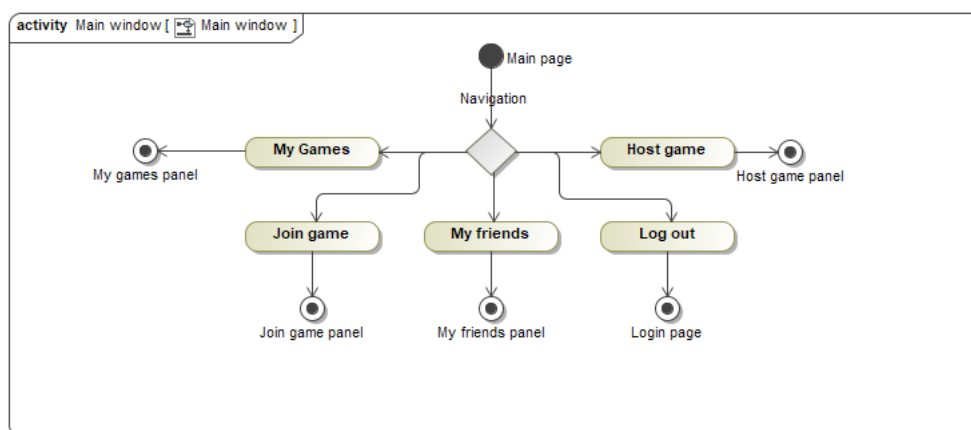
## Prisijungimo/Registracijos procesų veiklos diagrama :



16 pav. Autentifikacijos veiklos diagrama

Kiekvienas programėlės naudotojas patenka į prisijungimo langą, kuriame gali pasirinkti vieną iš galimų prisijungimo būdų - naudojantis Facebook API arba paprasta vartotojo paskyra. Pasirinkus Facebook autentifikaciją, programėlė nuskaito vartotojo identifikacinius duomenis ir ši veikla pabaigiama. Pasirinkus prisijungimą naudojantis paprasta paskyra atsiranda dvi galimybės - prisijungti arba užsiregistruoti. Pasirinkus registraciją vartotojas suveda savo duomenis, jeigu pastarieji teisingi - jis užregistruojamas ir gali prisijungti. Jei duomenys klaidingi - vartotojas vėl patenka į tą patį registracijos langą. Norėdamas prisijungti vartotojas turi įvesti savo teisingus prisijungimo duomenis, tuomet iššoka CAPTCHA patvirtinimo paveikslėlis, skirtas prevencijai nuo kenkėjiško tipo skriptų ir programų, imituojančių žmogų. Jeigu CAPTCHA klaidingas - vartotojas blokuojamas 5 minutėms (tiek laiko negali prisijungti siekiant taupyti sistemos resursus, išvengti DOS/DDOS atakų). Suvedus teisingą CAPTCHA kodą patenkama į sistemą ir ši veikla užbaigiama.

## 5.2. Pagrindinis navigacijos langas



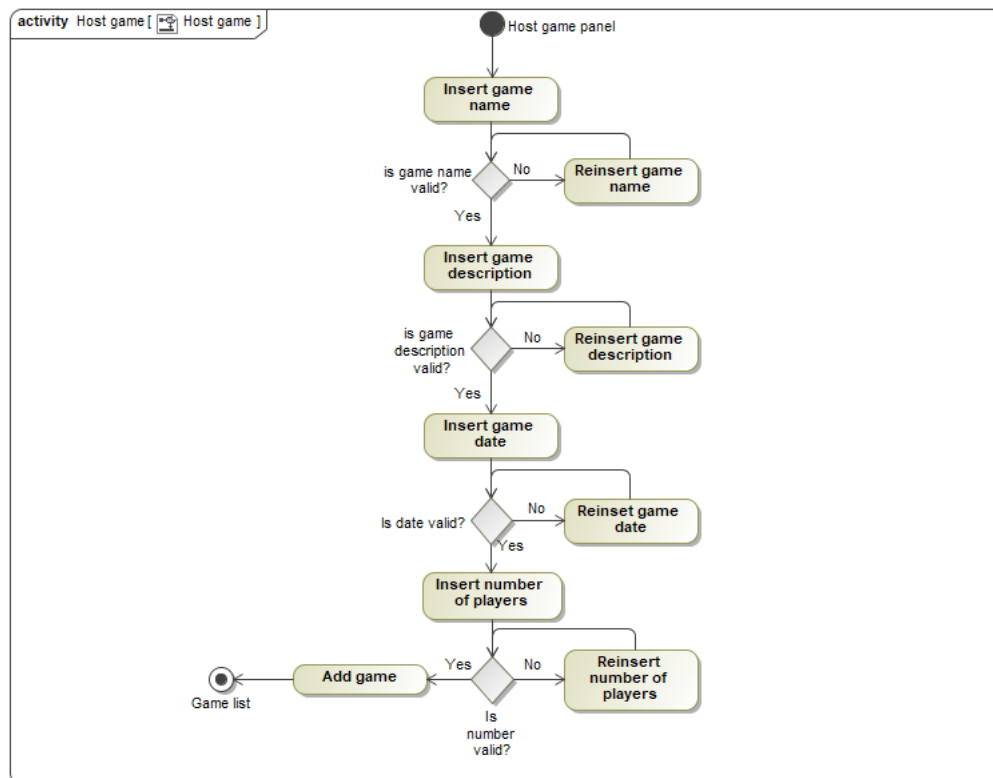
17 pav. Pagrindinio navigacijos lango veiklos diagrama

**Pagrindiniame navigacijos lange vartotojas paspaudęs atitinkamus mygtukus gali:**

- Atidaryti žaidimo sukūrimo langą.
- Atidaryti prisijungimo prie žaidimo langą.
- Atidaryti draugų sąrašo langą.
- Atidaryti mano žaidimų langą
- Atsijungti

### 5.3. Žaidimo sukūrimo langas

Žaidimo sukūrimo veiklos diagrama :



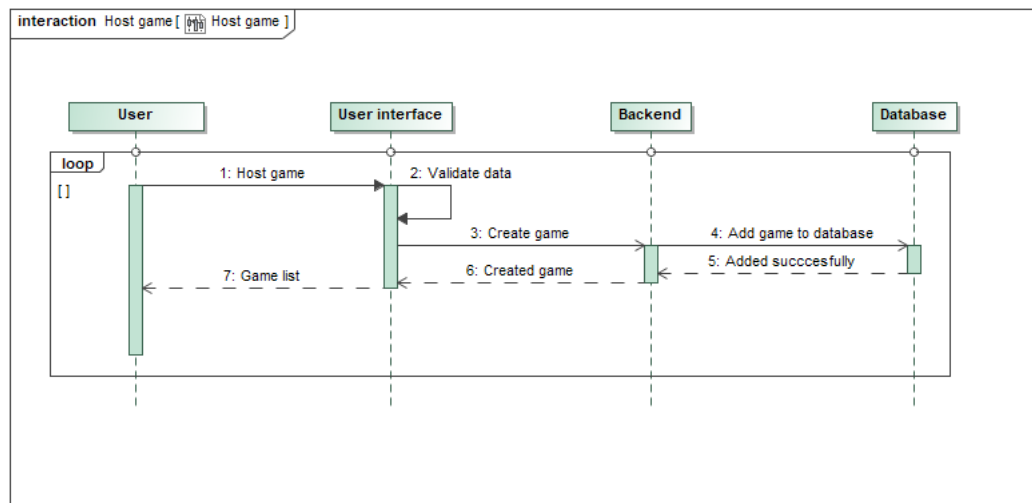
18 pav. Žaidimo sukūrimo veiklos diagrama

Žaidimo sukūrimo lange vartotojas turi įvesti :

- Žaidimo pavadinimą.
- Žaidimo aprašymą.
- Žaidimo datą.
- Žaidėjų skaičių

Įvedimo metu tikrinama ar duomenys įvesti leistinu formatu. Jei formatas netinkamas, vartotojas turi pataisyti atitinkamus duomenų laukus. Tinkamai užpildžius laukus, leidžiama pridėti žaidimą.

## Žaidimo sukūrimo procesų sekų diagrama :

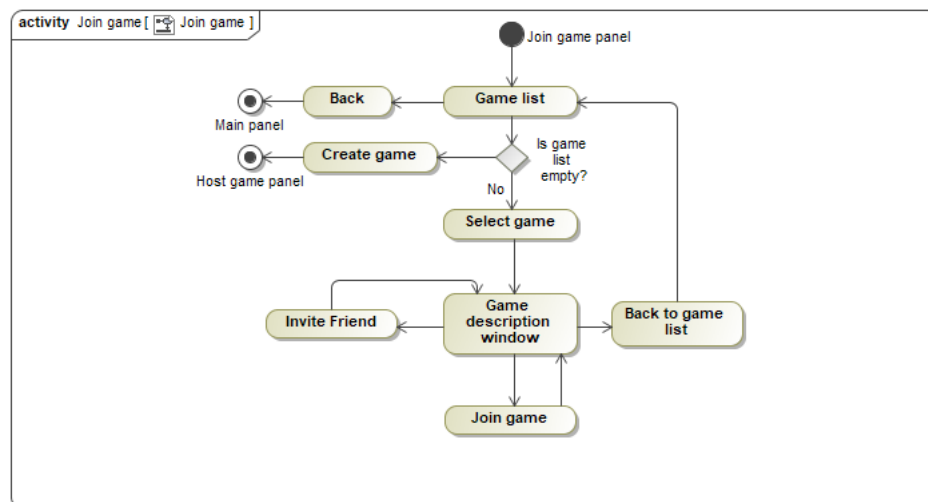


19 pav. Žaidimo sukūrimo procesų sekų diagrama

Vartotojas norėdamas sukurti žaidimą įveda žaidimo duomenis. Įvedimo metu tikrinama ar duomenys įvesti leistinu formatu. Jei duomenys tinkami, tada jie JSON formatu išsiunčiami loginiam procesui, kuris sukuria žaidimą ir įrašo jį į duomenų bazę.

## 5.4. Prisijungimo prie egzistuojančio žaidimo langas

### Prisijungimo prie egzistuojančio žaidimo veiklos diagrama :

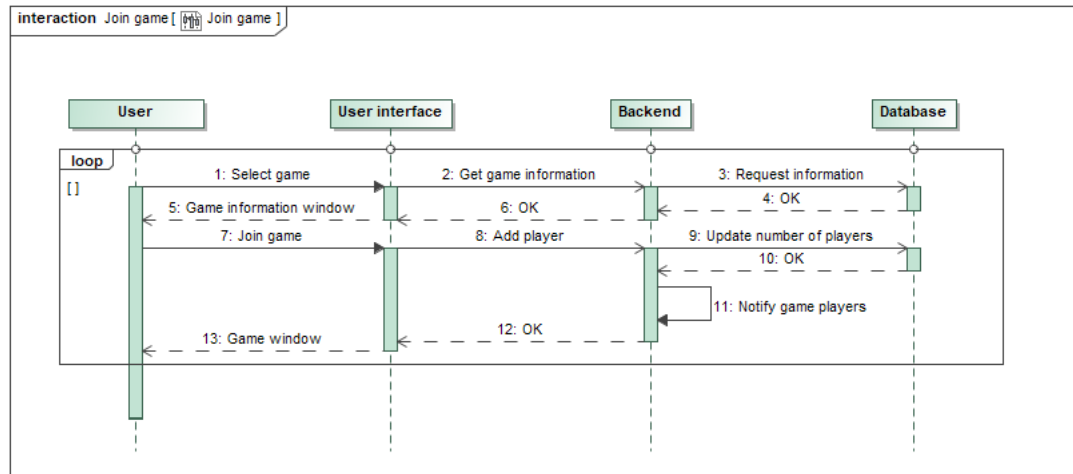


20 pav. Prisijungimo prie egzistuojančio žaidimo veiklos diagrama

Iš pradžių vartotojui atidaromas egzistuojančių žaidimų sąrašas. Čia vartotojas gali grįžti pagrindinį navigacijos langą arba tęsti žaidimo pasirinkimą. Jei nėra žaidimų prie kurių būtų galima prisijungti, vartotojui pasiūloma sukurti naują žaidimą. Vartotojui sutikus jis nukreipiamas į žaidimo sukūrimo langą. Jei žaidimų sąrašas nėra tučias, tai vartotojas gali pasirinkti žaidimą prie kurio norėtų prisijungti. Pasirinkus atidaromas pasirinkto žaidimo aprašymo langas. Čia vartotojas

gali pakviesti draugus kartu žaisti pasirinktą žaidimą, pats prisijungti prie žaidimo arba grįžti atgal į žaidimų sąrašą ir pasirinkti kitą žaidimą.

### Prisijungimo prie egzistuojančio žaidimo sekų diagrama :

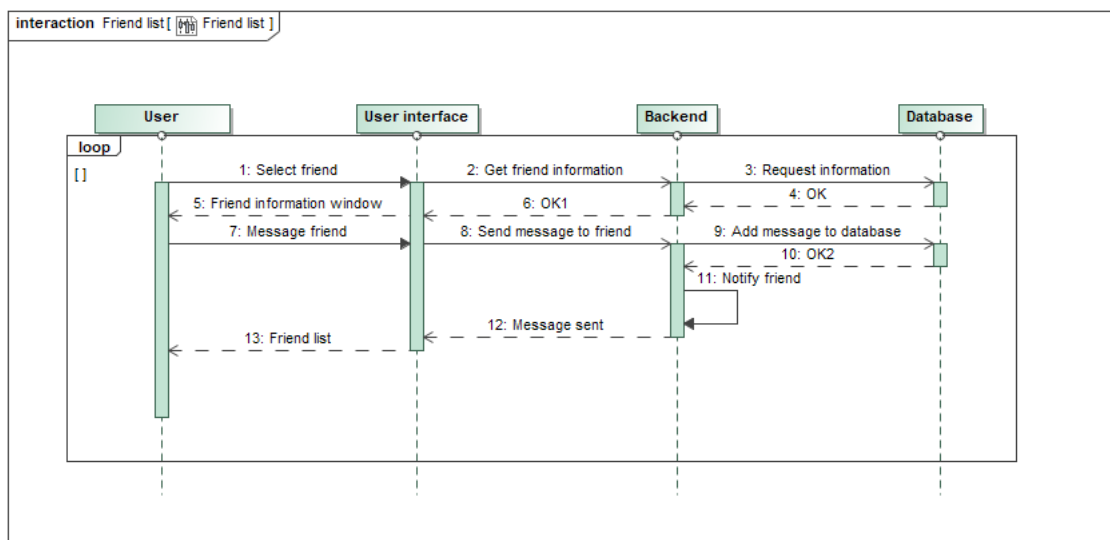


21 pav. Prisijungimo prie egzistuojančio žaidimo procesų sekų diagrama

Žaidėjui pasirinkus žaidimą iš sąrašo, atidaromas žaidimo informacijos langas. Informaciją apie žaidimą suteikia logikos procesas, kuris nusiuntęs užklausa į duomenų bazę, gauna papildomą informaciją apie pasirinktą žaidimą. Vartotojui nusprendus prisijungti prie žaidimo, nusiunčiama žinutė atnaujinti prisijungusių žaidėjų skaičių duomenų bazėje, taip pat visiems tame žaidime užsiregistravusiems žaidėjams išsiunčiamas pranešimas apie naują prisijungusį žaidėją.

## 5.5. Draugų sąrašo langas

### Draugų sąrašo lango sekų diagrama :

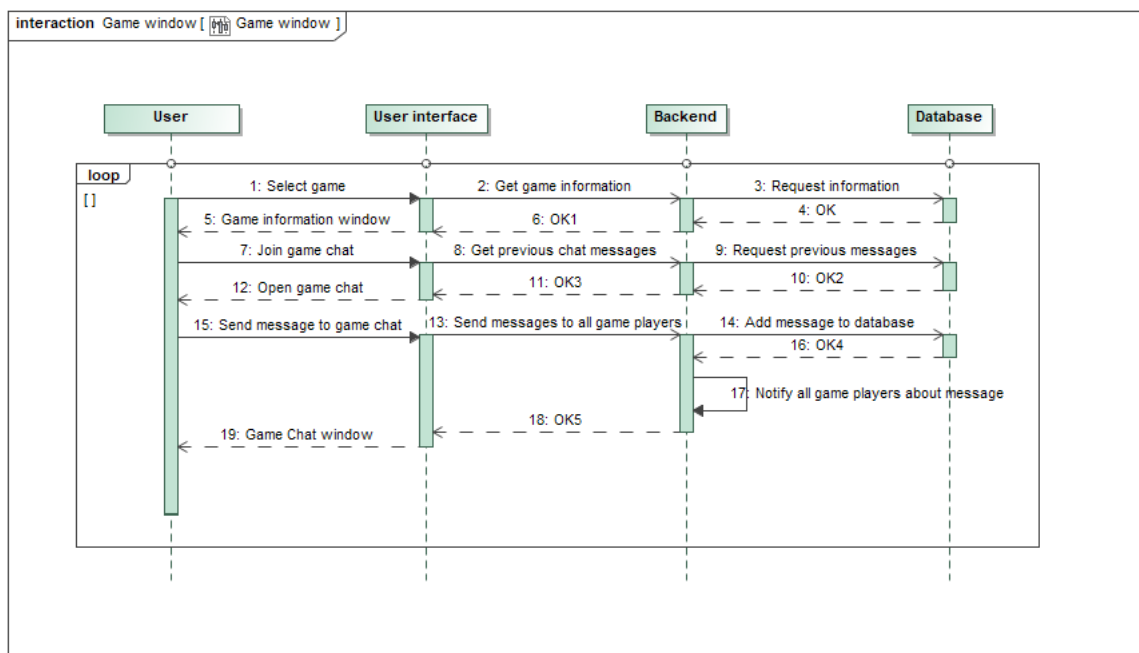


22 pav. Draugų sąrašo lango sekų diagrama

Vartotojui pasirinkus draugą iš draugų sąrašo, paprašoma loginio proceso gauti daugiau duomenų apie draugą iš duomenų bazės. Tuomet atidaromas langas su daugiau duomenų apie draugą. Paspaudus siųsti žinutę logikos procesas išsaugo žinutę duomenų bazėje ir praneša draugui apie gautą žinutę.

## 5.6. Žaidimo langas

Žaidimo lango sekų diagrama :



23 pav. Žaidimo lango sekų diagrama

Vartotojui pasirinkus žaidimą iš žaidimų sąrašo, paprašoma loginio proceso gauti daugiau duomenų apie žaidimą iš duomenų bazės. Tuomet atidaromas langas su daugiau duomenų apie žaidimą. Paspaudus siųsti žinutę logikos procesas paprašo žaidimo susirašinėjimo istorijos iš duomenų bazės. Tada atidaromas pokalbių langas, kur vartotojas gali išsiųsti žinutę visiems prie žaidimo prisijungusiems žaidėjams. Paspaudus siųsti žinutę logikos procesas išsaugo žinutę duomenų bazėje ir praneša žaidėjams apie gautą žinutę.



## Literatūros sąrašas

- Doc. dr. K. Petrausko Programų Sistemų Inžinerijos kurso konspektai
- UML dokumentacija [https://www.tutorialspoint.com/uml/uml\\_2\\_overview.htm](https://www.tutorialspoint.com/uml/uml_2_overview.htm)
- OMG UML v.2.5 Dokumentacija diagramoms, žymėjimui
- Informacija apie išdėstymo tinkle (Deployment) diagramas, pavyzdžiai <http://www.uml-diagrams.org/deployment-diagrams-overview.html>