

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

# Projektų valdymo sistema

Laboratorinis darbas

Atliko: 2 kurso 2 grupės studentai:

Vilius Minkevičius,

Edvinas Šmita,

Teodoras Šaulys,

Tomas Mikna,

Manfredas Šiurkus.

Projektų vadovas: Doc., dr. Karolis Petrauskas

## ANOTACIJA

Darbo tikslas – išsiaiškinti programų sistemų inžinerijos teikiamą naudą analizuojant verslo problemas ir kuriant pabaigtus programinės įrangos produktus užsakovams. Darbe yra sprendžiamas klausimas, kaip sukurti paprastą bei intuityvią projektų valdymo sistemą. Šio dokumento tikslas yra apžvelgti visas šio projekto detales, iki projekto įgyvendinimo versle kylančias problemas, jų sprendimo būdus.

# Turinys

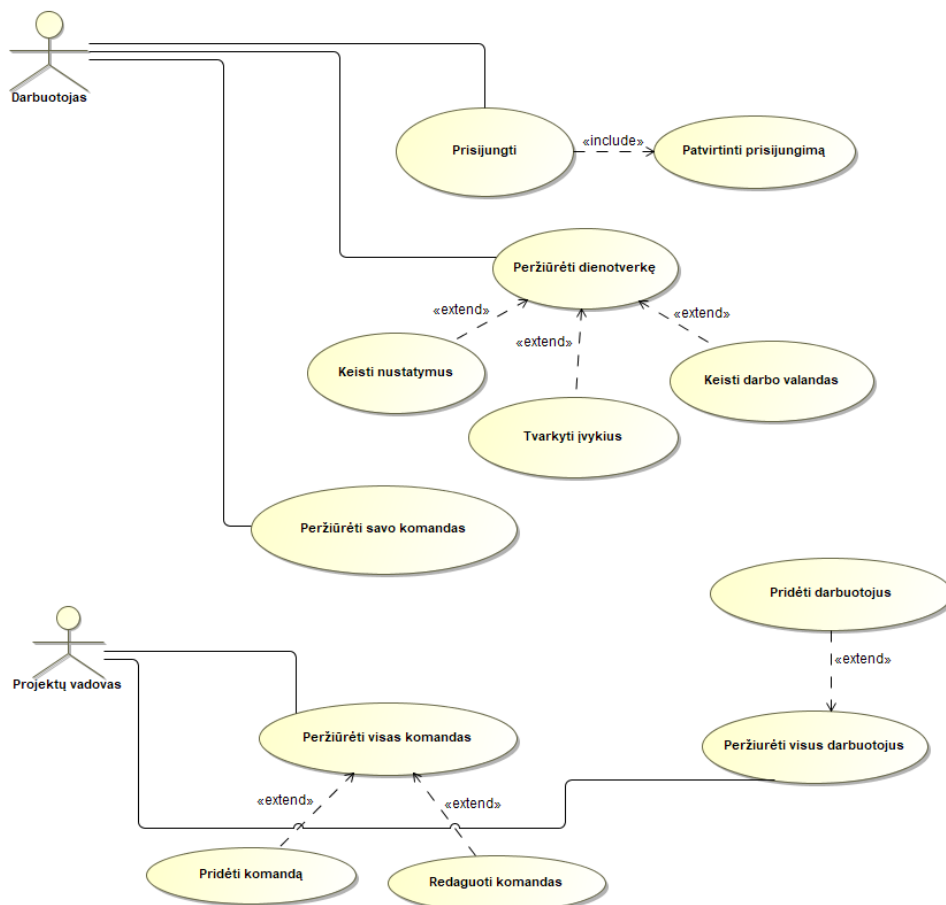
ĮVADAS .....	4
Užduotys (angl. Use case view).....	5
Struktūrinis programų sistemos modelis (angl. Logical view) .....	7
Dinaminis programų sistemos modelis (angl. Process view) .....	9
Bendrasis sistemos procesas .....	9
Bendri vartotojų procesai .....	10
Prisijungimo procesas .....	10
Prisijungimo patikrinimo procesas.....	10
Projektų vadovo procesai .....	11
Prašymų peržiūros procesas .....	11
Komandos sandaros keitimo procesas .....	12
Projekto kūrimo procesas: .....	12
Darbuotojo pridėjimo į sistemą procesas: .....	12
Darbuotojo procesai .....	13
Darbo žymėjimo procesas.....	13
Darbotvarkės keitimo procesas .....	13
Programų sistemos komponentai (angl. Development view) .....	14
Sekų diagrama.....	14
Komponentų diagrama .....	15
Komponentų išskirstymas tinkle (angl. Physical view).....	16
TERMINŲ ŽODYNAS .....	17
LITERATŪROS SĄRAŠAS.....	18
PRIEDAI .....	19

# IVADAS

Projektų valdymo sistemos architektūra pateikiama naudojant 4+1 pjūvių modelį. Dokumentą sudaro šios esminės dalys: užduotys, struktūrinis programų sistemos modelis, dinaminis programų sistemos modelis, programų sistemos komponentai ir komponentų išskirstymas tinkle (dokumento pabaigoje yra pridėta pjūvių modelio schema). Kiekviena dalis atvaizduoja programų sistemą ar jos dalį vis iš kito architektūros požiūrio taško.

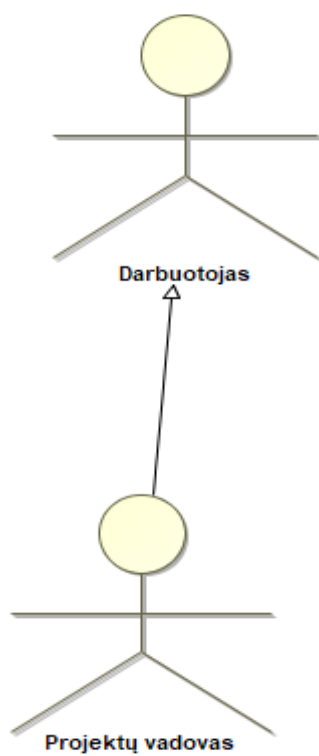
## Užduotys (angl. Use case view)

Šio skyriaus paskirtis yra apibrėžti pagrindinius programų sistemos tikslus (užduotis), kas parodo kryptį programų sistemos architektūrai. Tam naudojame UML naudojimo atvejų diagramą.



pav. 1 užduočių diagrama

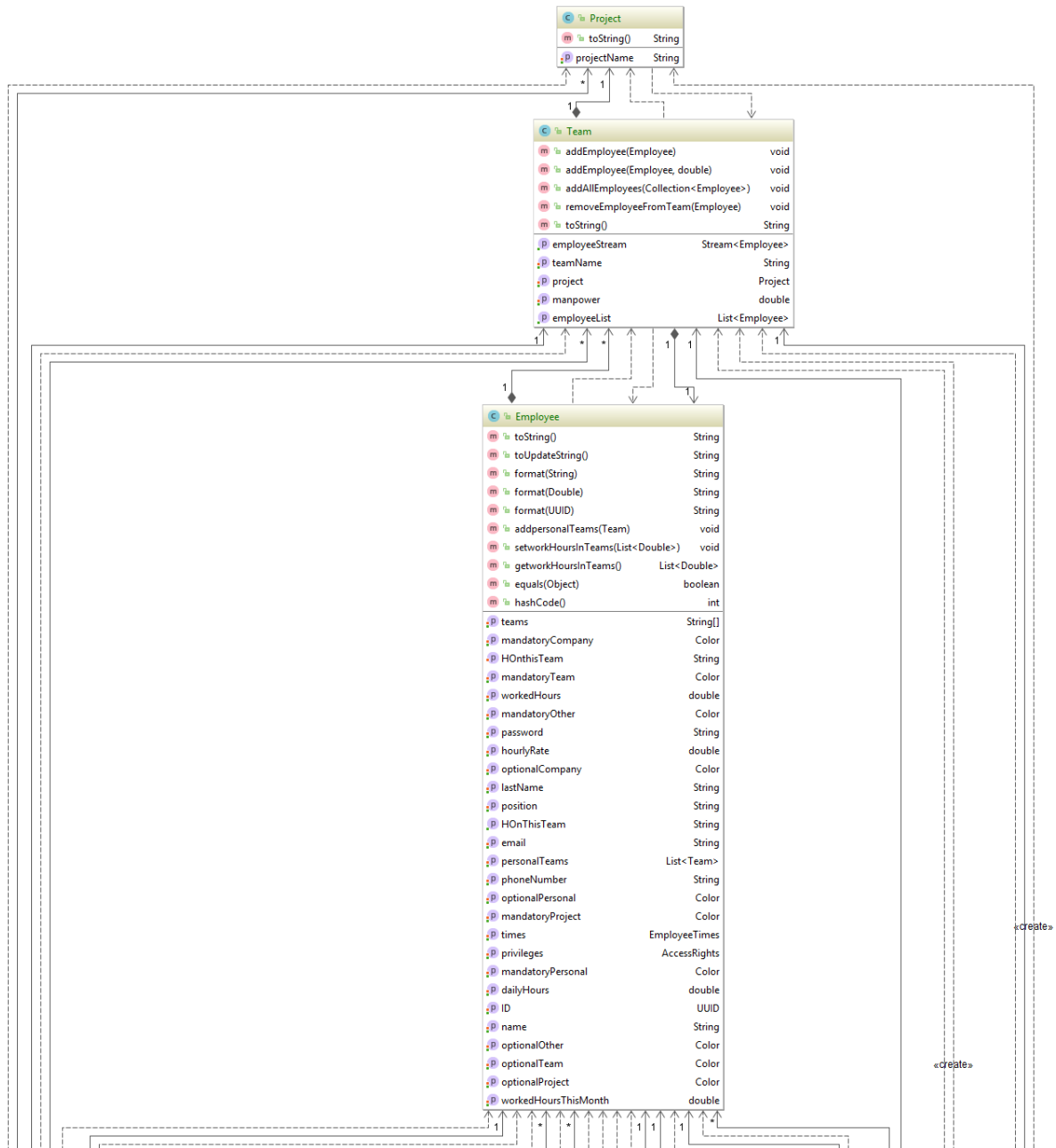
Naudojimo atvejų diagramoje matyti pagrindiniai dalykai, kuriuos naudotojui leidžia atlikti mūsų programų sistema. Taip pat, diagramoje matyti skirtingiems naudotojams prieinamos galimybės. Atskiroje, agentų hierarchijos, diagramoje matome ir vartotojų tarpusavio sąsajas. Užduočių įgyvendinimo scenarijai pavaizduoti tolimesniuose pjūviuose.



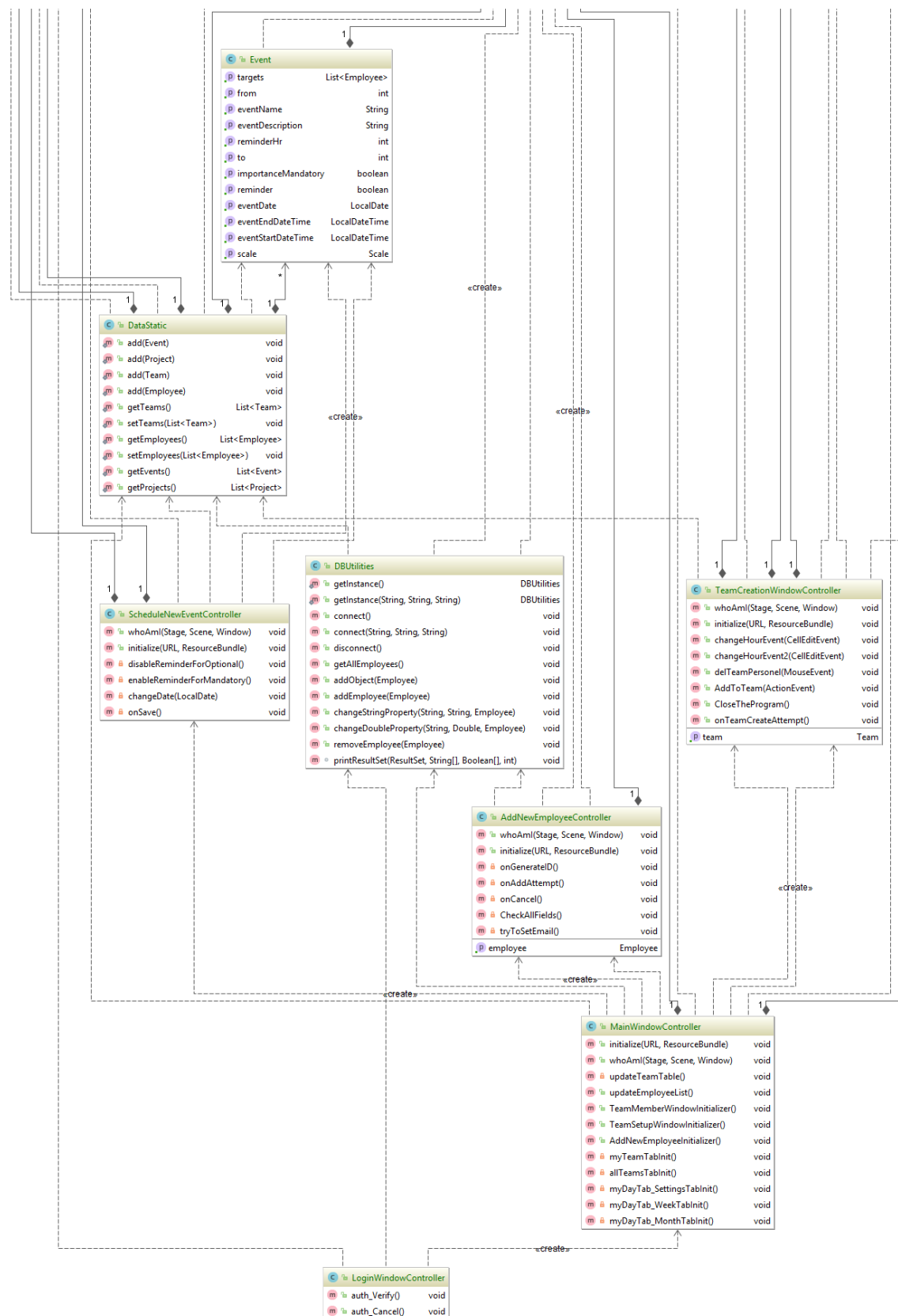
*pav. 2 agentų hierarchija*

Projektų vadovas gali veikti ir darbuotojo rolėje. Tai reiškia, jis irgi gali žymėti savo darbo progresą, žiūrėti tvarkaraštį bei įvykius kaip paprastas darbuotojas.

## Struktūrinis programų sistemos modelis (angl. Logical view)



pav. 3 Klasių diagramos pirma pusė



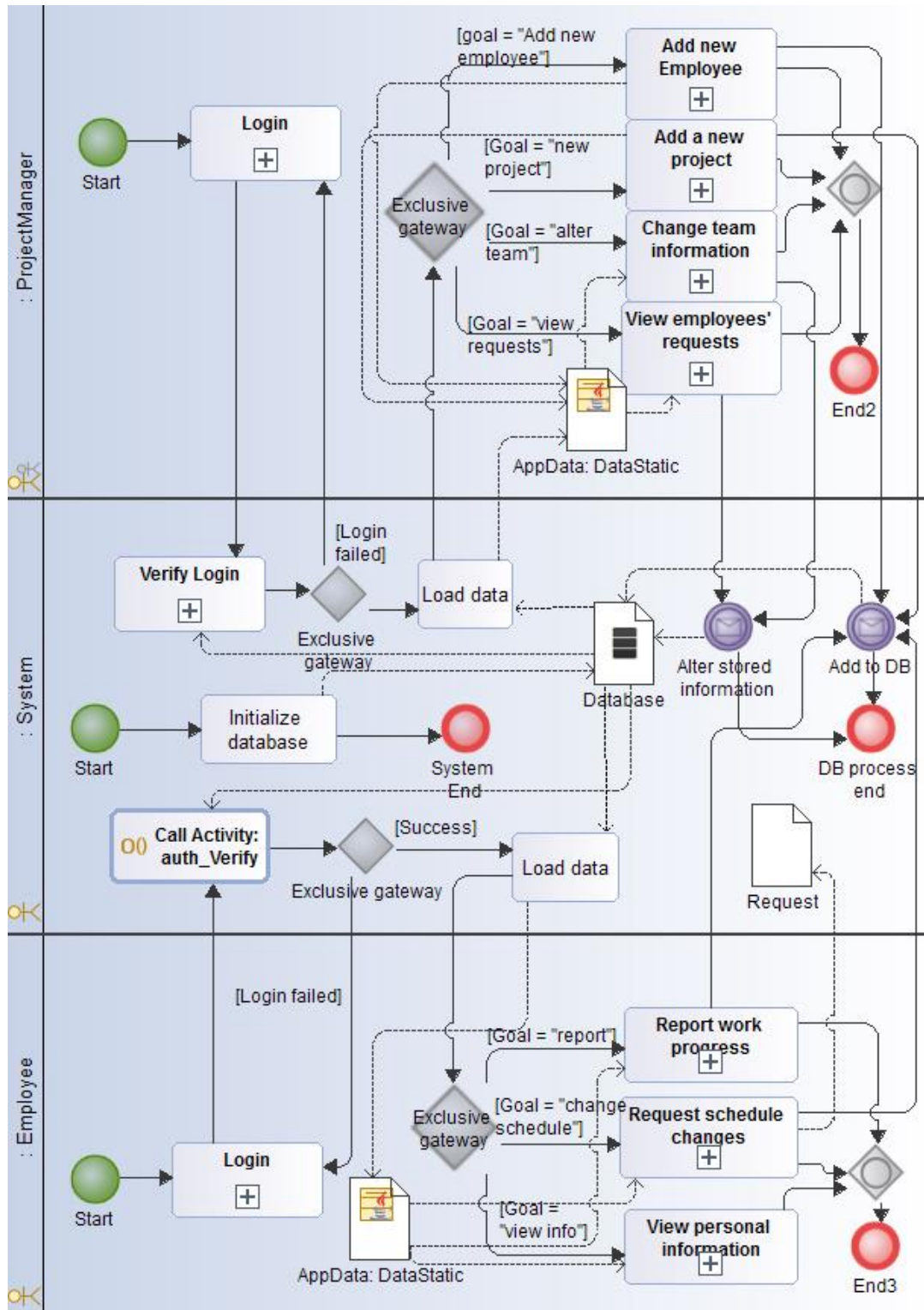
pav. 4 klasių diagramos antra pusė

Programų sistemos UML klasių diagrama, kurioje matosi svarbiausios klasės ir tarp jų egzistuojantys ryšiai bei jų tipai.



# Dinaminis programų sistemos modelis (angl. Process view)

## Bendras sistemos procesas

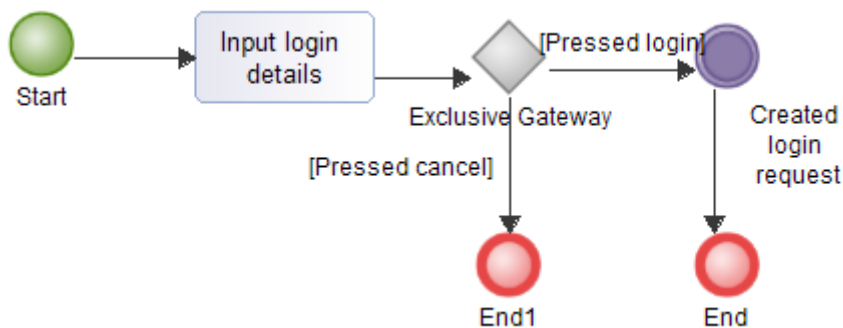


pav. 5 sistemos procesų diagrama

Pagrindinė procesų diagrama rodo, kaip sąveikauja projektų vadovas, darbuotojai ir projektų valdymo sistema. Darbuotojai bei projektų vadovai darbą su sistema pradeda prisijungdami (login). Prisijungimo duomenis įvedus ir paspaudus jungtis sistemoje yra patikrinimas prisijungimo prašymas. Netinkamai suvedus duomenis vartotojas yra prašomas bandyti prisijungti iš naujo, tinkamai – atveriamas pagrindinis programos langas, kuris atitinka vartotojo vaidmenį. Projektų vadovas mato projektų valdymo grafinę sąsają, o darbuotojai – projekto vykdytojų. Prisijungęs projektų vadovas gali kurti projektus, skirti darbuotojus į komandą, pridėti darbuotojus į sistemą bei peržiūrėti darbuotojų prašymus. Pridėjus ar pakeitus informaciją, pokyčiai yra išsaugomi duomenų bazėje. Darbuotojai gali peržiūrėti savo tvarkaraštį, žymėti darbo progresą ir sudaryti tvarkaraščio pakeitimo prašymus. Atlikęs veiklą vartotojas gali daryti kitą savo rolei tinkamą veiklą. Darbą baigęs vartotojas uždaro aplikaciją.

## Bendri vartotojų procesai

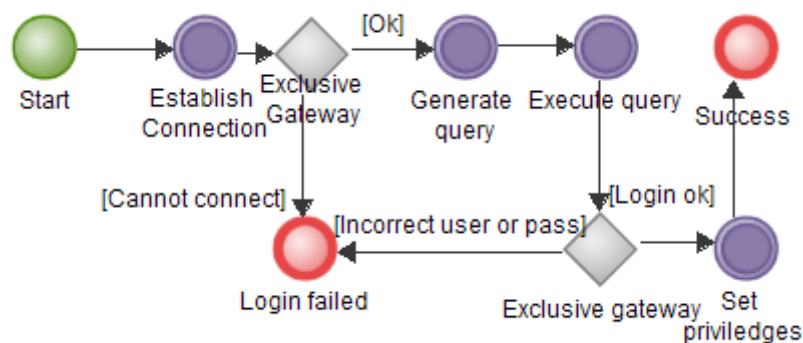
### Prisijungimo procesas



pav. 6 prisijungimo proceso diagrama

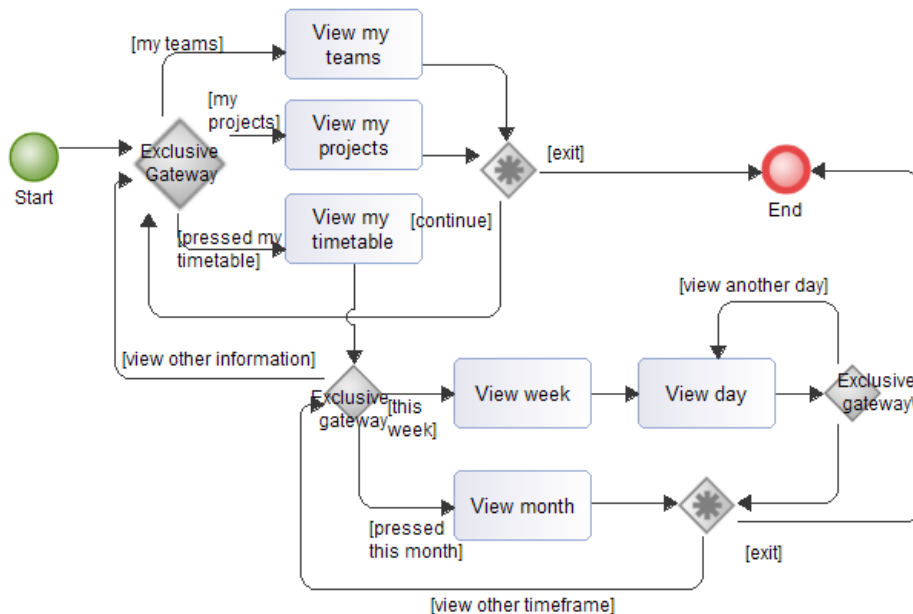
Prisijungimo lange vartotojas įveda savo prisijungimo duomenis. Paspaudus „login“ mygtuką programa išsiunčia prisijungimo užklausą į serverį. Priešingu atveju programa baigia darbą.

### Prisijungimo patikrinimo procesas



pav. 7 prisijungimo patikrinimo proceso diagrama

Kai vartotojas bando prisijungti, aplikacija siunčia užklausą į duomenų bazę, kurios rezultatas nulemia, ar pavyko prisijungti. Jei nepavyko prisijungti, prisijungimo procesas kartojamas.

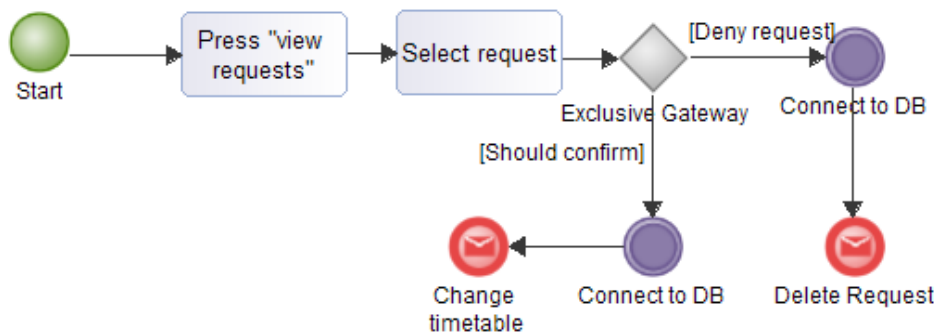


pav. 8 informacijos peržiūros proceso diagrama

Prisijungęs vartotojas turi galimybę peržiūrėti informaciją (savo komandas, tvarkaraštį, projektus).

## Projektų vadovo procesai

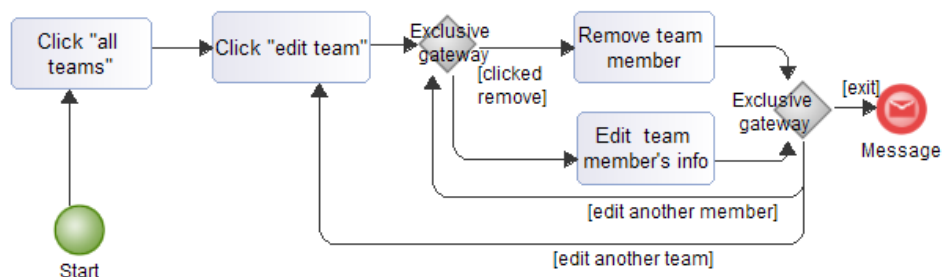
### Prašymų peržiūros procesas



pav. 9 prašymų peržiūros proceso diagrama

Projekto vadovas prisijungęs gali peržiūrėti savo pavaldinių prašymus keisti tvarkaraščius. Visi prašymai yra sudaryti, jei turi reikiamą valandų kiekį. Paspaudęs mygtuką prie prašymo gali jį atšaukti arba priimti. Priėmus prašymą bus pakeičiamas darbuotojo tvarkaraštis.

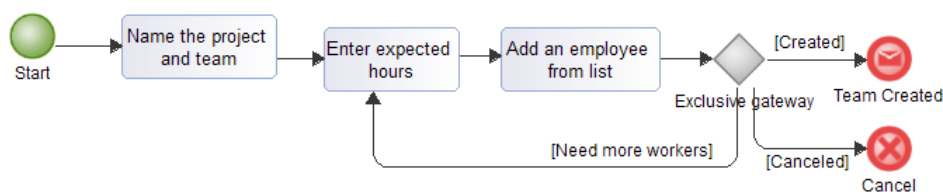
## Komandos sandaros keitimo procesas



pav. 10 komandos redagavimo proceso diagrama

Projektų vadovas gali peržiūrėti visas komandas. Taip pat gali jas keisti. Gali pridėti ir išimti darbuotoją norimą arba galimą kiekį kartų. Taip pat, užfiksavęs komandos pakeitimus gali keisti sekančia komandą arba baigti darbą.

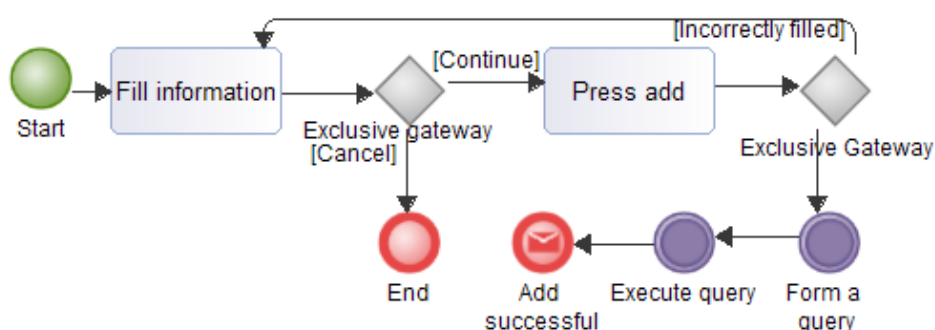
## Projekto kūrimo procesas:



pav. 11 projekto kūrimo proceso diagrama

Projektų vadovas turi galimybę sukurti projektą. Pridėti numatytą vykdymo laiką ir paskirti projektui darbuotojų.

## Darbuotojo pridėjimo į sistemą procesas:

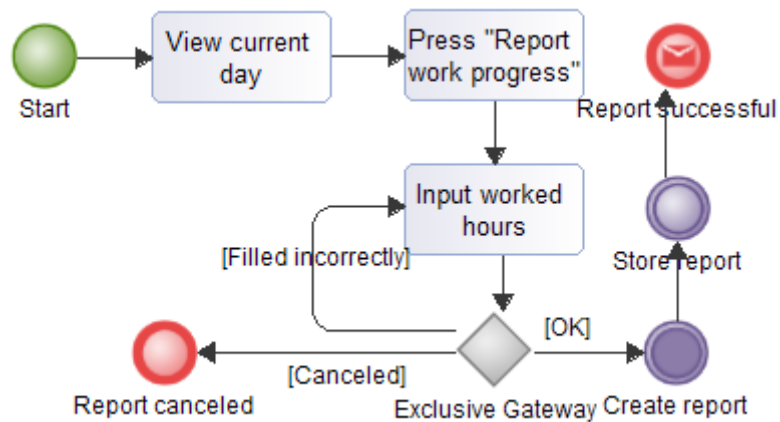


pav. 12 darbuotojo pridėjimo proceso diagrama

Projektų vadovas gali “įdarbinti” žmogų. Suvedama informacija ir paspaudus mygtuką “add” naujas darbuotojas pridedamas į duomenų bazę.

## Darbuotojo procesai

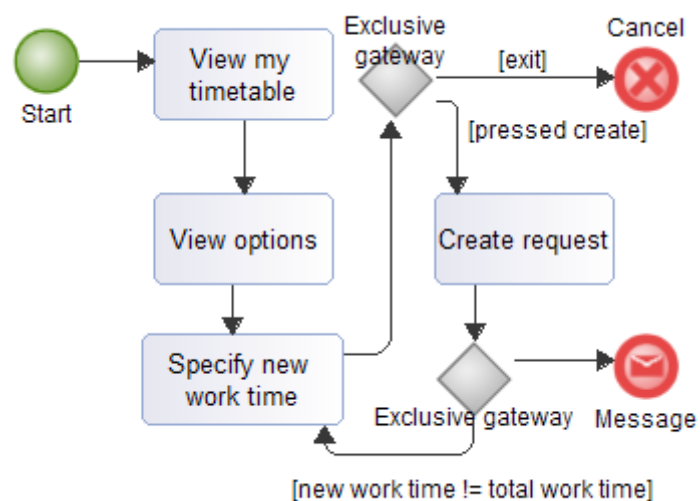
### Darbo žymėjimo procesas



pav. 13 darbo žymėjimo proceso diagrama

Kiekvieną dieną darbuotojas turi galimybę papildyti savo darbo progresą. Jis gali keisti darbo valandas norimą kiekį kartų tol, kol patvirtina arba atšaukia formą.

### Darbotvarkės keitimo procesas

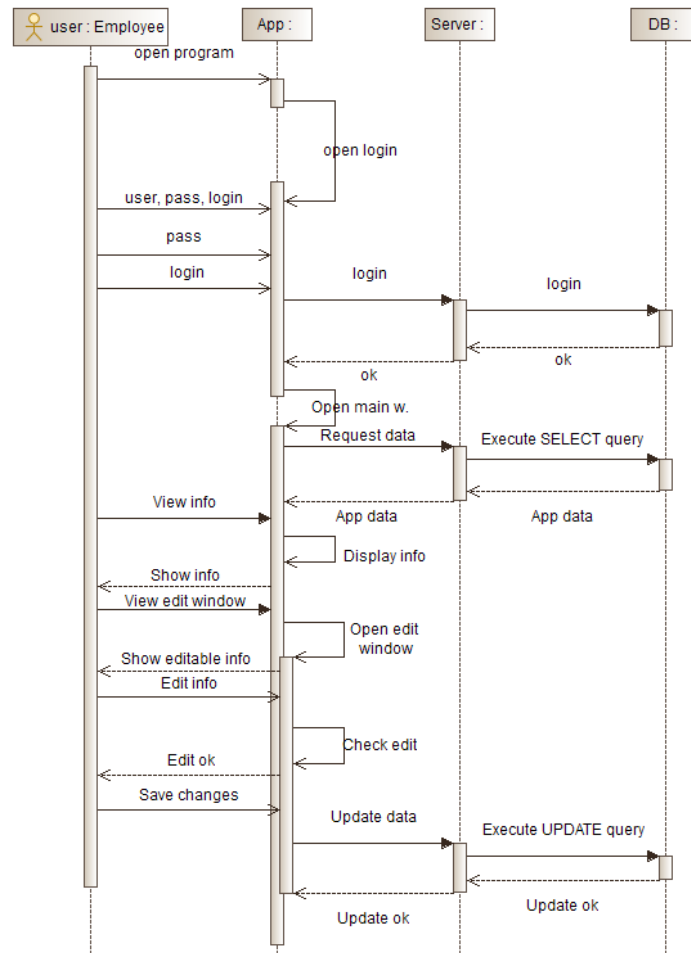


pav. 14 darbotvarkės keitimo proceso diagrama

Prisijungęs vartotojas turi galimybę peržiūrėti savo darbo laiką. Gali peržiūrėti savo tvarkaraščio nustatymus. Taip pat gali keisti, pridėti darbo laiką. Jei užklausa užpildyta netinkamai - programa prašo vartotojo pataisyti dokumentą. Tinkamai užpildytas prašymas išsaugomas ir nusiunčiamas patvirtinimui.

# Programų sistemos komponentai (angl. Development view)

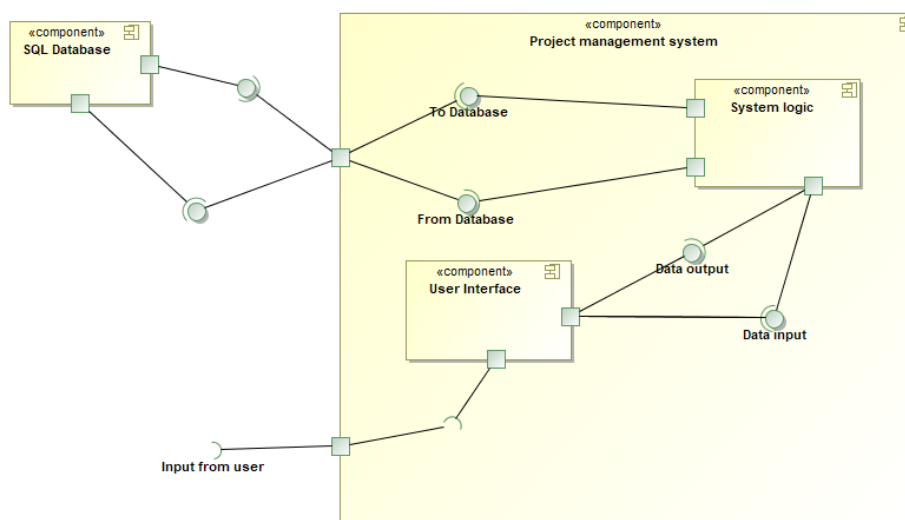
## Sekų diagrama



pav. 15 Sistemos sekų diagrama

Sekų diagramoje matome, kaip agentas bei sistemos komponentai sąveikauja vykdant pagrindines užduotis. Vartotojas tiesiogiai naudoja aplikaciją ir per ją - netiesiogiai su serveriu bei duomenų baze. Kai vartotojas bando jungtis, aplikacija prašo serverį patikrinti, ar prisijungimo duomenys teisingi. Sėkmingai prisijungus, aplikacija kreipiasi į serverį dėl vartotojui aktualios informacijos. Iš duomenų bazės per serverį gauta informacija patalpina aplikacijoje ir rodoma vartotojui šios paprašius. Vykdant pakeitimus, aplikacija patikrina, ar išlaikytas tinkamas formatas. Jei taip yra, leidžiama išsaugoti pokyčius ir vykdomas informacijos atnaujinimas sistemoje.

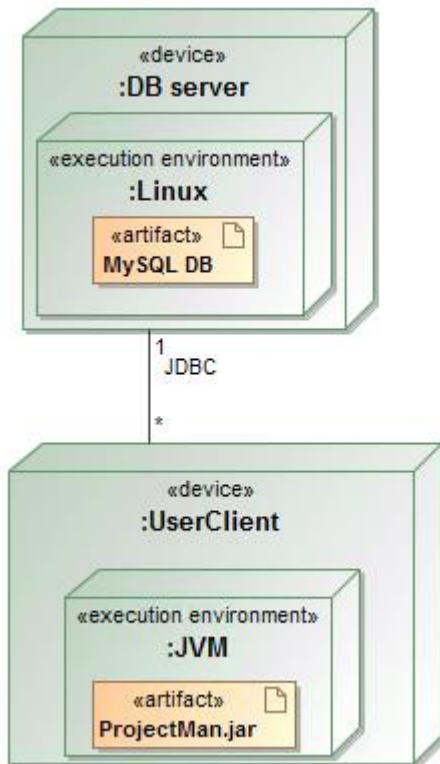
## Komponentų diagrama



pav. 16 komponentų diagrama

Komponentų diagrama nurodo ryšius tarp sistemos komponentų ir dviejų subkomponentų. Šie du sistemos subkomponentai yra grafinė aplinka ir sistemos logika. Grafinė aplinka, žinoma, atsakinga už duomenų suvedimą, tuo tarpu sistemos logika vykdo duomenų apykaitą tarp duomenų bazės ir sistemos.

## Komponentų išskirstymas tinkle (angl. Physical view)



pav. 17 Tinklo komponentų diagrama

Šioje komponentų diagramoje pavaizduota suplanuotas išdėstymas tinkle:

- Egzistuoja vienas MySQL duomenų bazės serveris, kuriame laikomi visi sistemoje naudojami duomenys. Šis serveris komunikuoja su potencialiai keliais kliento įrenginiais vienu metu HTTP protokolu, naudojant SQL užklausas.
- Kliento įrenginyje su Java pagalba yra paleista mūsų sistema, kurioje galima manipuluoti ir sinchronizuoti su duomenų baze visus mūsų sistemoje naudojamus duomenis.



# TERMINŲ ŽODYNAS

Svarbiausi mūsų vartojami terminai:

**Programų sistema, programa** - Šiame dokumente aptariama Projektų valdymo sistema.

**Darbuotojas** (employee) - Bet kuris įmonėje dirbantis, su tam tikru projektu susijęs programų sistemos naudotojas.

**Projektų vadovas** (manager) - atskiras darbuotojo atvejis, asmuo atsakingas už tam tikro projekto vystymą.

**Vartotojas** (user) – Asmuo besinaudojantis programų sistema, abstraktesnis darbuotojo atitikmuo.

**Duomenų bazė** (data base) - vieta kur saugomi duomenys (darbuotojų sąrašai, komandos ir t.t.)

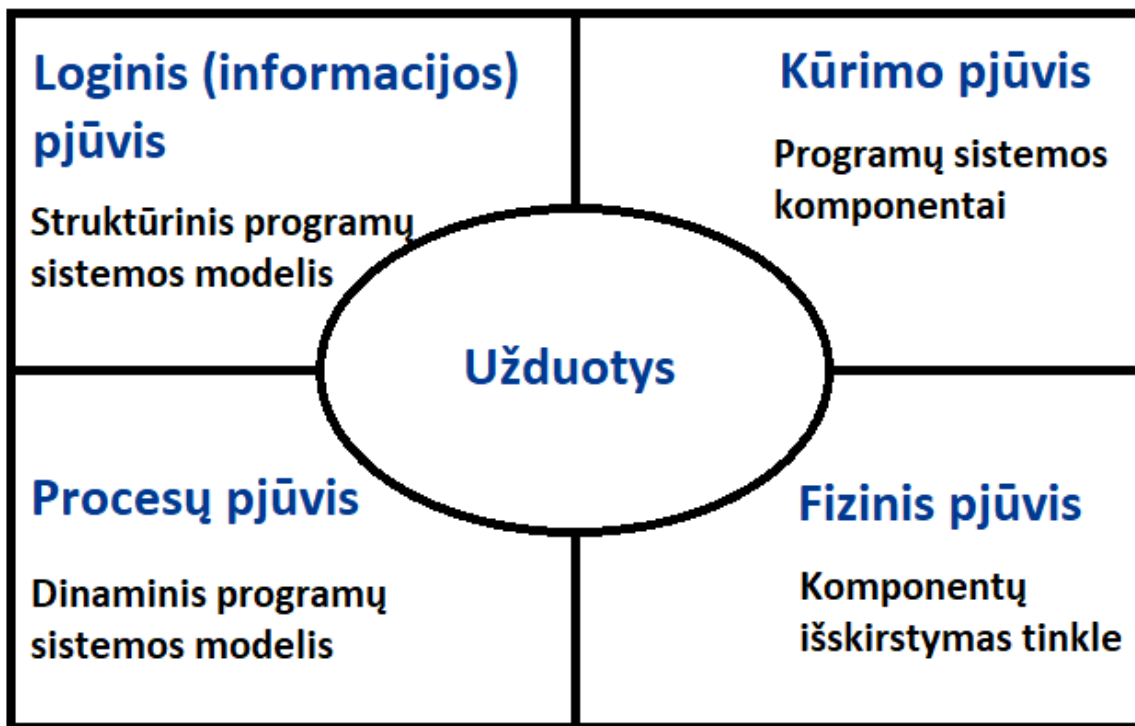
**Serveris** (server) - specialios paskirties programa ir kompiuteris (nebūtinai), skirta programų sistemos aptarnavimui.

# LITERATŪROS SĄRAŠAS

Paskaitų užrašai

## PRIEDAI

4+1 pjūvių modelis, kuriuo remiantis pateikiama programų sistemos architektūra



*pav. 18 Programų sistemos pjūviai*