|  |  |
| --- | --- |
|  |  |

**Laika kontroles programma brīvpiekļuves datoriem “TimeStation”**

**Programmatūras projektējuma apraksts**

TS.PPA.A1.2024

**2024**

**Laika kontroles programma brīvpiekļuves datoriem “TimeStation”**

**Programmatūras projektējuma apraksts**

TS.PPA.A1.2024

|  |  |
| --- | --- |
| **Izstrādāja:** | R. Parakevičs |

|  |  |
| --- | --- |
| **Kvalitātes pārvaldnieks:** | R. Parakevičs |

Izmaiņu lapa

Iepriekšējais dokuments: iepriekšējā dokumenta identifikators

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Izmaiņu apraksts** | **Vieta iepriekšējā dokumentā** | **Izmaiņu pamatojums** |
|  |  |  |  |
|  |  |  |  |

Saturs

[1. Ievads 1-1](#_Toc184754220)

[1.1. Nolūks 1-1](#_Toc184754221)

[1.2. Darbības sfēra 1-1](#_Toc184754222)

[1.3. Definīcijas, akronīmi un saīsinājumi 1-1](#_Toc184754223)

[1.4. Saistība ar citiem dokumentiem 1-1](#_Toc184754224)

[1.5. Dokumenta raksturojums 1-1](#_Toc184754225)

[2. TimeStation vispārīgs apraksts 2-2](#_Toc184754226)

[1. Attēls – Lietotāja saskarne 2-2](#_Toc184754227)

[2.1. TimeStation sadarbība ar citiem produktiem un sistēmām 2-2](#_Toc184754228)

[2.2. Projekta izstrādes ierobežojumi 2-2](#_Toc184754229)

[2.2.1. Projekta izstrādes vide 2-2](#_Toc184754230)

[2.2.2. Projekta versiju uzturēšana caur GitHub 2-3](#_Toc184754231)

[2.2.1. Projekta iestatījumu saglabāšana 2-3](#_Toc184754232)

[2.2.2. Bibliotēka WPF Extended Toolkit 2-3](#_Toc184754233)

[2.2.3. Bibliotēka WPF Material Design 2-3](#_Toc184754234)

[2.2.4. Bibliotēka WPF MahApps 2-3](#_Toc184754235)

[3. TimeStation arhitektūra 3-4](#_Toc184754236)

[3.1. Funkcionālā skatupunkta elementi 3-4](#_Toc184754237)

[3.1.1. Savienojuma loģika 3-4](#_Toc184754238)

[2. Diagramma - komunikācijas serverim 3-4](#_Toc184754239)

[3.1.2. Laika pievienošanas loģika 3-5](#_Toc184754240)

[3. Diagramma – Laika pieslēgums 3-5](#_Toc184754241)

[3.1.3. Ziņas nosūtīšanas loģika 3-5](#_Toc184754242)

[3.1.4. Naudas skaitīšanas loģika 3-6](#_Toc184754243)

[3.1.5. Datoru izslēgšanas loģika 3-6](#_Toc184754244)

[3.2. Informācijas skatupunkta elementi 3-6](#_Toc184754245)

[3.2.1. Klientu identifikācija 3-6](#_Toc184754246)

[3.2.2. Klientu nosaukumi 3-6](#_Toc184754247)

[3.2.3. Naudas koeficients 3-7](#_Toc184754248)

[3.2.4. Brīdinājuma ziņas teksts 3-7](#_Toc184754249)

[3.3. Izvietošana skatupunkta elementi 3-7](#_Toc184754250)

[3.3.1. Tīkla izkārtojums 3-7](#_Toc184754251)

[3.3.2. Klienta palaišana 3-7](#_Toc184754252)

[4. TimeStation funkcionālais projektējums 4-7](#_Toc184754253)

[4.1. [Funkcijas nosaukums] 4-8](#_Toc184754254)

[4.1.1. [Funkcijas nosaukums] būtība 4-8](#_Toc184754255)

[4.1.2. [Funkcijas nosaukums] loģika 4-8](#_Toc184754256)

[*5.* [Funkcijas nosaukums] lietotāja saskarnes projektējums 5-8](#_Toc184754257)

[5.1. XAML 5-8](#_Toc184754258)

# Ievads

“TimeStation” ir programmu pāris, kas dod lietotajiem iespēju savienot kopā vienā tīklā vairākus klienta datorus un vienu galveno administratīvo datoru.

Dokuments sastāv no vispārēja apraksta par programmu, kas tālāk noved pie katras sistēmas daļas apraksta, funkciju detalizētas izpētes, saskarnes apraksta un modeļa apraksta.

## Nolūks

Programmas paredz, ka ar administratīvo datoru, lietotājs spēj ātri un efektīvi kontrolēt piekļuvi un izmantošanas laiku noteiktajiem klienta datoriem. Veidojot programmas prasību specifikācijas dokumentu, tika izteikti vairāki procesi un darbības, kurām jātiek realizētām iekš programmām. Šis dokuments paredz šo realizāciju aprakstu veidošanu un izkārtošanu lasītājiem saprotamā veidā, lai jebkurš programmas lasītājs, gan izstrādātājs, vai gala lietotājs spētu orientēties kodā un saprast kā gala produkts strādā

## Darbības sfēra

Dokuments galvenokārt ir paredzēts projektētājiem un programmētajiem, jo šeit tiek dziļi un smalki aprakstītas funkcijas, to vajadzības un izveides kārtība. Šis apraksts ir jāizmanto lai veiksmīgi varētu veikt programmas izveidi un uzturēšanu.

## Definīcijas, akronīmi un saīsinājumi

|  |  |
| --- | --- |
| Termins | Skaidrojums |
| Administratīvā programma | Galvenā programma caur kuru ir iespējas veikt pieslēgto klientu datoru administrāciju. |
| Administrators | Lietotājs kuram ir pieeja pie administratīvā datora. |
| Lietotāja programma | Programma kas atrodas uz jebkura kontrolējama datora un sniedz administratīvās programmas funkcionalitāti. |
| Lietotājs | Jebkurš cilvēks kas izmanto brīvpieejas datoru. |
| Savienojums | Savienojums starp administratīvo un klienta datoru kas atbild par abpusējo kumunikāciju programmas darbībai. |
| Pieslēgums | Laika pieslēgums klienta datoram kas nosaka cik ilgi var izmantot noteikto datoru. |
| Brīdinājums | Nosūtāms teksts vai teikums, kurš parādās uz norādīto klienta datoru. |
| Bloķēt | Izslēgt piekļuvi pie izvēlētā klienta datora. |
| Atbloķēt/Pieslēgt/Aktīvs | Atļaut piekļuvi pie klienta datora, ieslēdzot tam izmantošanas laiku. |

## Saistība ar citiem dokumentiem

* TS.PPS.A2.2024.docx – Programmatūras prasību specifikācijas dokuments.
* GitHub repozitorijs, kas satur versiju un izmaiņu vēsturi un pirmkodu: <https://github.com/Rudis-P/Datoru_Kontrole>
* Changelog.docx – Trasējamību nolūkiem veidota tabula priekš veiktajām izmaiņām.

## Dokumenta raksturojums

Katrai no uzskaitītajam funkcijām vai galvenajām daļām ir apraksts, ievades un izvades dati, shēma, saistība ar citam funkcijām un dažiem ir pievienots grafiks/diagramma. Dokuments satur skaidrojumus un aprakstus katrai sistēmas sastāvdaļai, caur kuriem ir vizuāli vai tekstuāli apzīmēta funkciju darbība.

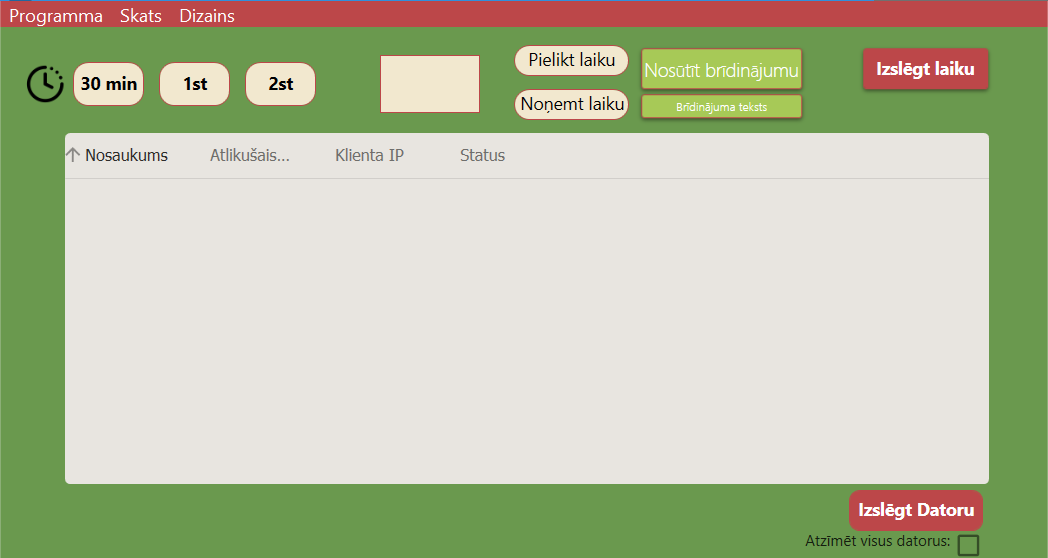
# TimeStation vispārīgs apraksts

Programmas sniedz lietotajam iespēju kontrolēt vienu vai vairākus savienotus datorus, nosakot cilvēku pieeju tiem. Programmas strādā uz Klienta-Servera bāzes, kur administratīvā programma uzvedas kā serveris un ļauj lietotajiem noteikt laika pieslēgumu, caur kuru klienta programma vadās.

**Administratīvā programma** ir galvenā programma, kurai ir pieejama lietotāja saskarne, kas ļauj izmantot visas piedāvātās funkcijas. Caur šo galveno programmu var redzēt visus savienotos datorus un ir iespējams tos tieši un ātri kontrolēt vai bloķēt.

**Klienta programma** tiek instalēta uz jebkura datora, kuru vēlas kontrolēt. Šai programmai ir divi galvenie stāvokļi: Bloķēts un Atbloķēts. Bloķētajā stāvoklī datoru nav paredzēts izmantot un tas neļauj lietotājam tajā neko darīt, ja nav pieslēgts laiks. Ja ir pieslēgts laiks, atbloķētajā stāvoklī klients datoru var izmantot un ekrāna augšas vidū vienmēr tiek rādīts atlikušais laiks. Kad laiks beidzas, vai tas tiek manuāli izslēgts, dators atgriežas bloķētajā stāvoklī.

Tikai administratīvajai programmai ir izmantojama lietotāja saskarne:



1. Attēls – Lietotāja saskarne

## TimeStation sadarbība ar citiem produktiem un sistēmām

Programma atbalsta tikai Windows operētājsistēmas, Windows 7 un jaunākas, kurām ir aktīvs pieslēgums lokālam interneta tīklam. Caur šo tīklu tiek veikti savienojumi un komunikācijas starp programmām.

Pašas programmas darbībā nav iekļautas citas sistēmas, kuras būtu atsevišķi jāinstalē lai nodrošinātu darbību.

## Projekta izstrādes ierobežojumi

Šeit ir izteikti ierobežojumi, kuri attiecas uz veidojamās programmas izstrādes procesu, precīzāk par izmantojamajām tehnoloģijām un bibliotēkām.

### Projekta izstrādes vide

Programmas tiek veidotas iekš Microsoft **Visual Studio 2022 IDE** izstrādes vides. Caur šo vidi tiek izmantota WPF programmas izstrādne.

**Kāpēc –** Programmu izveidei tika izvelētaWPF tipa programma, jo tā sniedz ērtu veidu kā komplektēt viegli izmantojamas programmas Windows operētājsistēmu datoriem. WPF sniegtais vizuālais dizaineris, caur XAML, dod iespēju izveidot attiecīgu lietotāja saskarni, kas palīdz programmētajiem savienot lietotāja darbības ar pašu kodu.

### Projekta versiju uzturēšana caur GitHub

Paralēli programmas izveidei, tiek aktīvi izmantota tīklā bāzētais pakalpojums GitHub, kas ļauj uzturēt programmas kodu repozitorijā, kas ļauj aktīvi sekot līdzi izmaiņām.

**Kāpēc –** GitHub ir bezmaksas un tas sniedz iespēju veikt izmaiņas un papildināt programmas no vairākiem datoriem, viegli centralizējot programmas pirmkodu. Papildus tam tas sniedz vieglu veidu kā sekot līdzi izmaiņām, uzlabojot funkciju trasējamību.

### Projekta iestatījumu saglabāšana

Programmas procesā lietotājs var veikt izvēles kuras ir nepieciešams saglabāt. Šis tiek realizēts ar Visual Studio iebūvēto projekta iestatījumu lapu.

**Kāpēc –** Sniedz iespēju vieglā un kopīgā vietā apkopot visus vajadzīgos mainīgos, kurus ir nepieciešams saglabāt, pasargājot tos no netīšas izdzēšanas vai nesaglabāšanas.

### Bibliotēka WPF Extended Toolkit

Šī bibliotēka nodrošina WPF programmu izstrādes procesam papildus iespējas un elementus lai pilnveidotu lietotāja pieredzi un padarītu izstrādāšanas procesu dziļāku.

**Kāpēc –** Šī bibliotēka tiek izmantota lai sniegtu dziļāku izstrādes procesu un papildinātu programmas elementus ar funkcijām, kā piemēram, krāsas izvēles elements priekš lietotajā saskarnes.

### Bibliotēka WPF Material Design

Šī bibliotēka nodrošina WPF programmu izstrādes procesam daudz vizuālo iespēju, kas papildina jau iekļautās WPF programmas elementu vizuālo dizainu.

**Kāpēc –** Izmantojot noklusētos elementus un to izskatu, programmas izskats var šķist garlaicīgs un prasts. Ar šo bibliotēku ir iespēja dziļāk pamainīt pamat elementu izskatu, jo pēc noklusējuma tiek iekļauti pamainīti dizaini elementi. Papildus

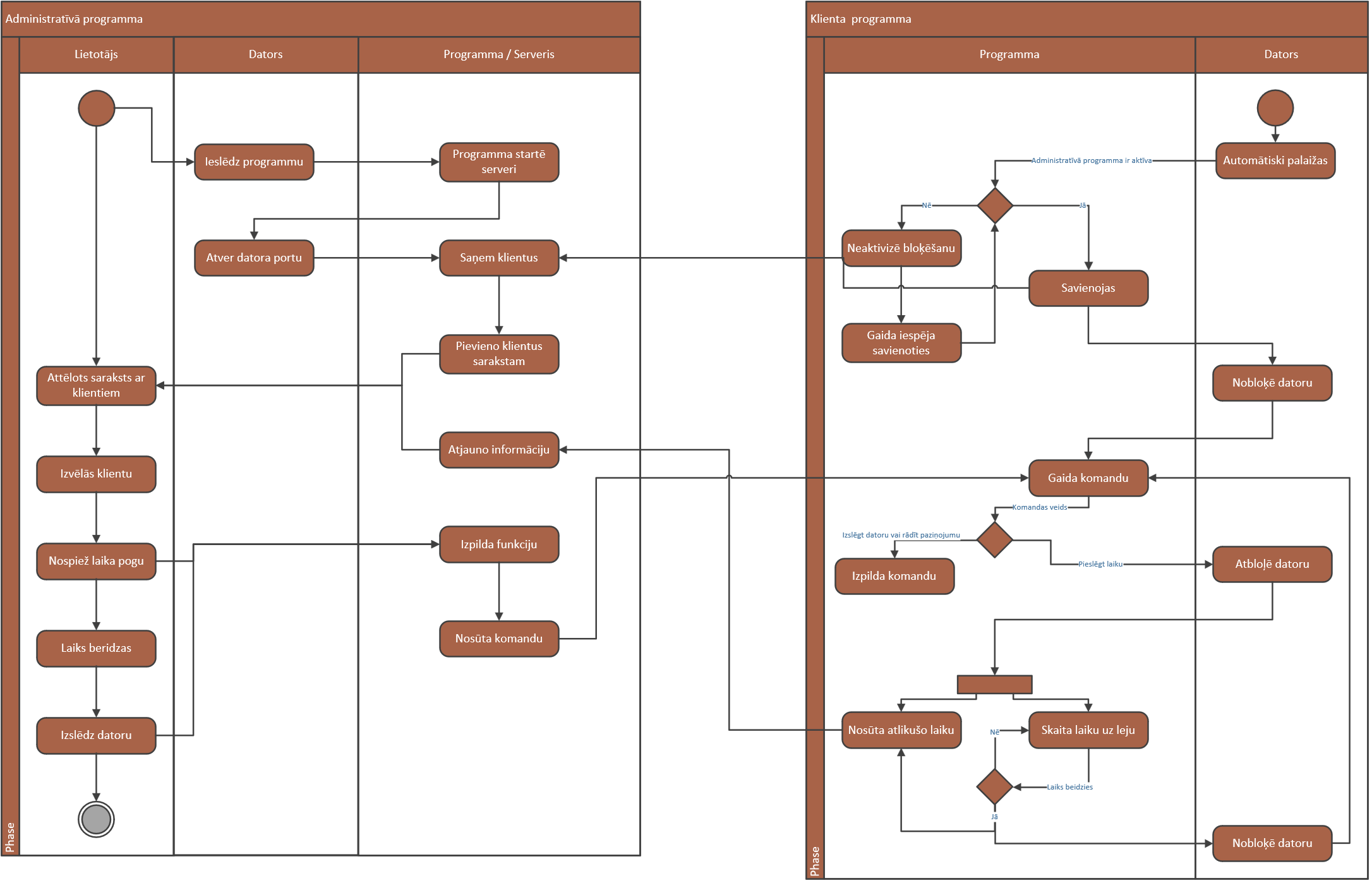
### Bibliotēka WPF MahApps

Šī bibliotēka nodrošina WPF programmu dizaina elementus, kas tematiski atbilst Windows dizaina stilam.

**Kāpēc –** Ar šo paplašinājumu ir iespēja izveidot vizuālus elementus, kas ir līdzīgi Windows videi, kā arī sniedz papildus izkārtojuma iespējas logiem un vairākus papildus navigācijas elementus. Ar šo paplašinājumu tiek papildināts pats loga dizains un elementu izkārtojums iekš tā, lai sniegtu modernu saskarni.

# TimeStation arhitektūra

Programmas arhitektūru ir iespējams aprakstīt ar skatupunktu palīdzību, sadalot visu programmu aprakstāmos gabalos, kur katrs satur ar sevi saistītus elementus un to aprakstus.



1. Diagramma – Aktivitātes diagramma

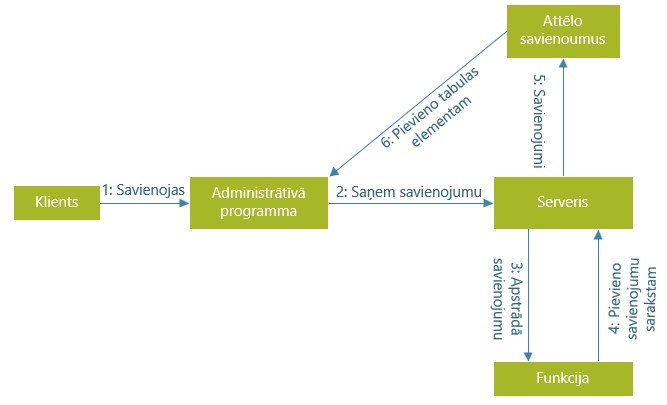
## Funkcionālā skatupunkta elementi

Programmas realizācijai ir svarīgi vairāki elementi, kas izpilda noteiktas funkcijas. Šīs funkcijas nodrošina programmu ar tās darbību, kā tas ir noteikts izstrādei. Katra programmas daļa, kas atbild par funkciju izpildi tiek uzskaitītas un aprakstītas, izsakot to būtību un vajadzību. Ņemot vērā ka abas programmas komunicē viena ar otru, regulāri sazinoties, katrs elements ir papildināts ar UML(Unified Modeling Language) komunikācijas diagrammu. Ar šo diagrammu ir viegli parādīt kā ziņas tiek nosūtītas un saņemtas un uz ko noteiktā ziņa attiecās.

### Savienojuma loģika

Funkcija attiecas uz programmas galveno funkcionālo prasību veidot savienojumus ar klienta datoriem kopīgā tīklā.

Savienojums strādā no administratīvās programmas atvērtā datora porta, kurš saņem savienojumus, pievienojot tos programmai. Attiecīgi caur šo savienojumu tiek nosūtītas komandas klienta programmai.



1. Diagramma - komunikācijas serverim

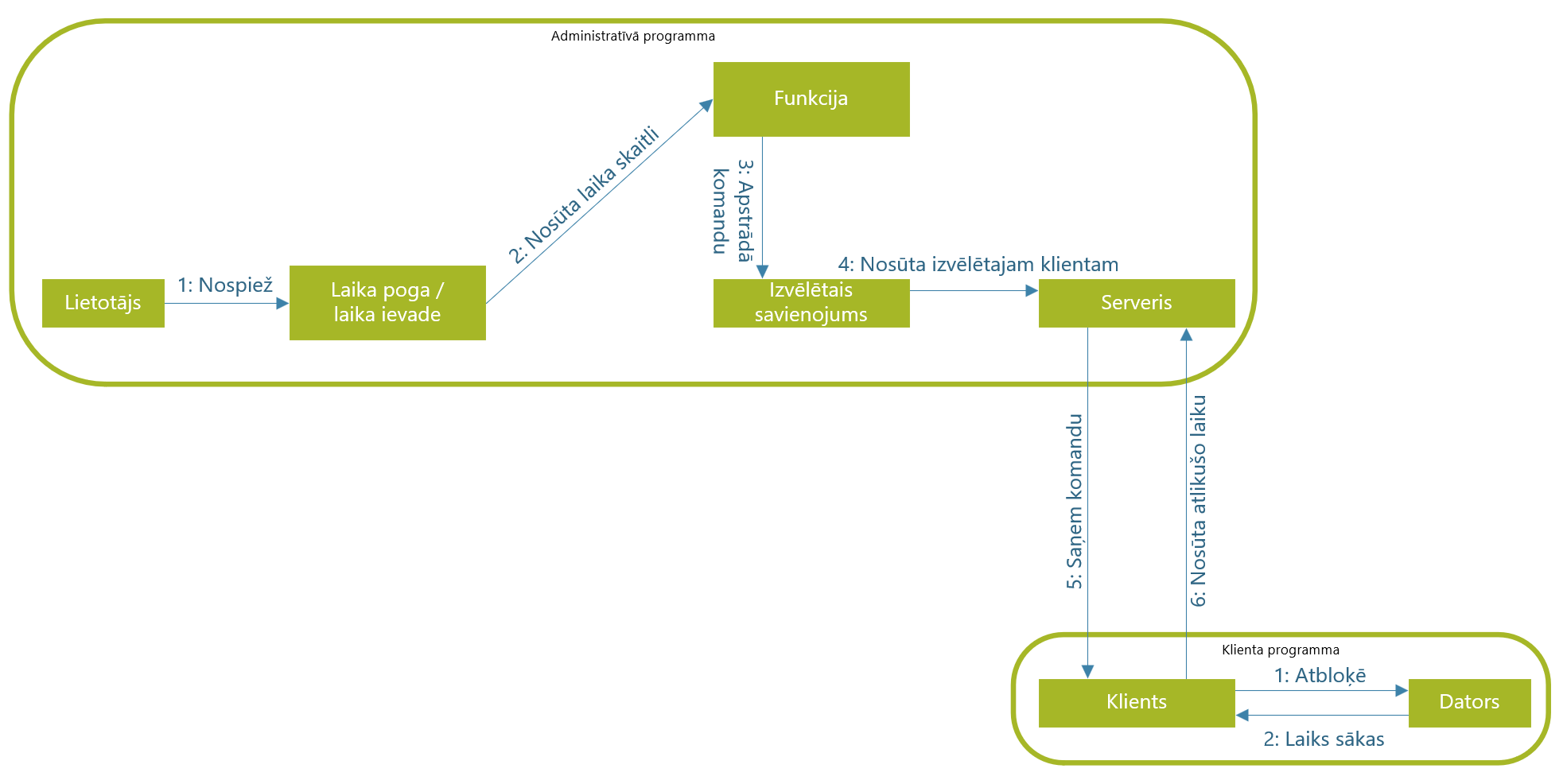
**Funkcionālie elementi:**

* Servera izveidošana – Tiek noteikts savienoju izveides punkts programmas kodā, kurš tiek aktivizēts palaišanas brīdī.
* Klientu pievienošana un noņemšana – Saņemtie savienojumi tiek apstrādāti un pievienoti sarakstam. Attiecīga ja savienojums ir pārtraukts vai pazūd, tad klienta ievads tiek izņemts.

**Saistītie saskarnes elementi** – Klientu parādīšanai tiek izmantots saskarnes tabulas elements “DataGrid”. Šajā tabulā tiek uzrādīti visi aktīvi savienotie klienti.

### Laika pievienošanas loģika

Līdz ko ir veiksmīgi izveidots savienojums, lietotājs ar šiem klientiem var veikt laika administrēšanu. Pievienoto laiku, atlikušo laiku un klienta stāvokli ir nepieciešams precīzi noteikt un attēlot, lai lietotājs varētu būt informēts par pašlaik notiekošo.



1. Diagramma – Laika pieslēgums

**Funkcionālie elementi:**

* Laika pievienošana – No lietotāja puses tiek noteikta laika vienība, kas tiek nosūtīta klientam caur savienojumu.
* Standarta laika pievienošanas pogas – Trīs pogas(30, 60, 120 minūtes), kuras piedāvā standarta laika pievienošanu, nenorādot laika skaitli pašam lietotājam.
* Laika noņemšana – Izmantojot to pašu pieeju kā pievienošanai, atlikušo laiku ir iespējam samazināt, ievadot laika skaitli ko noņemt no pašreizējā laika.
* Laika attēlošana – Vizuālajā vidē tiek attēlots atlikušais laiks, gan ar teksta rindu, gan ar krāsu palīdzību.
* Laika atjaunošana – Aktīvs reāllaika process, kurš atjauno atlikušo laiku. Klienta programma ir atbildīga par sava laika stāvokļa sūtīšanu uz serveri, kas tad ir attiecīgi apstrādāts administratīvajā programmā.

**Saistītie saskarnes elementi** – Trīs pogas, attiecīgajiem laikiem, kā arī lauks skaitļa ievadei priekš pievienošanas un noņemšanas.

### Ziņas nosūtīšanas loģika

Lietotājam ir pieejama iespēja nosūtīt personalizētu ziņu izvēlētajiem klientiem. Šī ziņa satur noklusējuma tekstu, bet lietotājs var to mainīt, lai sniegtu atbilstīgajam datoram, atbilstīgu paziņojumu. Ziņas nosūtīšana strādā tieši tāpat kā laika, tikai laika vietā tiek nosūtīts teksts. Uz klienta ekrāna šo ziņu ir nepieciešams apstiprināt nospiežot pogu, kas to aizver.

**Funkcionālie elementi:**

* Ziņas nosūtīšana – Nosūta ziņas tekstu izvēlētajam klientam. Teksts parādās klienta datorā kā logs pašā centrā, kurš pārklāj lielu ekrāna daļu.
* Teksta ievade ziņai – Zem nosūtīšanas pogas ir iespēja ievadīt tekstu kurš tiks nosūtīts. Šis aizstās noklusējuma tekstu, bet netiek saglabāts.

**Saistītie saskarnes elementi** – Administratīvajā programmā ir divas pogas, viena ziņas nosūtīšanai, otra atver teksta logu ziņas ievadīšanai. Un klientam viena, kas aizver ziņas logu.

### Naudas skaitīšanas loģika

Ja lietotājs izvēlās to ieslēgt, programma spēj skaitīt patērētu naudas summu, kura tiek parādīta uz ekrāna. Naudas summa tiek aprēķināta vadoties pēc formulas un lietotāja norādītā koeficienta.

**Funkcionālie elementi:**

* Naudas skaitītājs – Ieslēdzot funkciju tiek parādīts neliels logs programmas apakšdaļā, kur aktīvi tiek uzskaitīta naudas summa. Šī summa parādās tikai ja ir izvēlēts klients, jo katram klientam var būt sava summa, kas tiek individuāli saglabāta.
* Naudas koeficients – Tas ir skaitlis, kuru nosaka lietotājs. Pēc šī skaitļa tiek veikti aprēķini programmas kodā. Skaitli var mainīt lietotājs jebkurā brīdī.

**Saistītās saskarnes elementi** – Naudas skaitītāja summas skaitlis loga apakšdaļā. To var ieslēgt tikai ja naudas skaitīšanas funkcija ir ieslēgta programmai.

### Datoru izslēgšanas loģika

Jebkurā brīdī programmas lietotājam ir iespēja izslēgt vienu vai vairākus datorus nospiežot vienu pogu. Šis vienmēr izslēgs datoru, pat ja ir aktīvs laika savienojums.

**Funkcionālie elementi:**

* Datoru izslēgšana – Funkcija izpildīs izslēgšanas komandu, nosūtot attiecīgo komandu izvēlētajam klientam. Pēc pogas nospiešanas, klients uzreiz izpildīs to un izslēgsies.

**Saistītie saskarnes elementi** – Poga loga apakšdaļā.

## Informācijas skatupunkta elementi

Šajā sadaļā ir aprakstīts kā programma strādā ar tai pieejamajiem datiem. Programma aktīvi neapstrādā daudz datu, bet tās procesam ir svarīgi vairāki mainīgi vai vērtības kas ir aktuālas procesam.

### Klientu identifikācija

Katru reizi kad tiek veikts savienojums, tiek aktivizēta programmas klase, kurā ir noteikts kā izskatās savienojamais objekts. Šī klase tiks aprakstīta sīkāk nākamajā sadaļā, bet priekš informācijas skatupunkta vajag zināt trīs lietas: klienta datora IP adrese, nosaukums un status. Pēc šiem laukiem programma spēj veikt turpmāko darbību ar visām iesaistītajām programmām.

**Informācijas elementi:**

* IP adrese – Tiek piešķirta savienojuma brīdī, tā palīdz identificēt katru unikālo datoru kas ir savienojies.
* Nosaukums – Lietotāja ievadīts nosaukums, pēc noklusējuma tas ir tukšs.
* Status – Pašreizējais klienta stāvoklis, kas tiek atspoguļots saskarnē.

### Klientu nosaukumi

Kā tika minēts iepriekšējā punktā, klientiem ir iespējams piesaistīt lietotāja ievadītu nosaukumu. Caur saskarni ir iespējams, ar labo peles klikšķi, atvērt logu nosaukuma ievadei. Šis ievadītais nosaukums tiek atspoguļots saskarnē, rādot to pie savienotā klienta rindas tabulā. Šis nosaukums tiek saglabāts lokālā failā, kur katru reizi kad programma ir atvērta, no tā tiek nolasīts attiecīgais nosaukums priekš attiecīgā datora IP adreses.

**Informācijas elementi:**

* Nosaukums – Teksta mainīgais nosaukums katram individuālajam klienta savienojumam. Tiek saistīts ar IP adrese, lai varētu identificēt attiecīgo vārdu, attiecīgajam datoram.

### Naudas koeficients

Līdzko lietotājs ir izvēlējies par patērēto laiku sekot līdzi naudas summai, programma aprēķina šo summu balstoties uz formulu un koeficientu. Šo koeficientu nosaka lietotājs izmantojot programmas augšējo izvēles rindu. Mainot koeficientu, lietotājs var noteikt cik ātri summa aug.

**Informācijas elementi:**

* Koeficients – skaitļa mainīgais, kurš tiek izmantots formulā. To var izmainīt programmas laikā un formulas darbība netiks pārtraukta.

### Brīdinājuma ziņas teksts

Nosūtot brīdinājumu, tiks nosūtīts noklusējuma teksts “Lūdzu uzvedaties klusāk!”, bet ja lietotājs vēlās, tad ir iespēja nomainīt šo tekstu uz jebkādu citu izmantojot pogu brīdinājuma tekstam. Arī šeit atvērsies logs, kurā lietotājs var ievadīt teksta rindu, kura tiek saglabāta.

**Informācijas elementi:**

* Brīdinājuma teksts – teksta mainīgais, kuram ir noklusējuma vērtība.

## Izvietošana skatupunkta elementi

Programmas veiksmīgai darbībai ir nepieciešams pareizi to sastādīt, lai nerastos problēmas ikdienas darbā. Šis process nav sarežģīt, taču tas ir svarīgs lai programmas varētu strādāt kā tām vajadzētu.

### Tīkla izkārtojums

Ir ļoti svarīgi zināt ka programmas darbosies tikai tad, ja abas ir kopīgā tīklā. Līdz ko tiek palaistas programmas, tās veidos savienojumu balstoties uz lokālā tīkla saziņu.

**Izvietošanas elementi:**

* Kopīgais tīkls – Standarta interneta konfigurācija, kurā uz viena tīkla ir vairāki datori.

### Klienta palaišana

Klienta programma ir veidota lai tā būtu automātiska un nebūtu vajadzība mainīt neko tajā. Ņemot vērā ka klienta programmai neko nevar mainīt, ir nepieciešams pareizi uzstādīt to uz katra klienta datora. Tas ir sasniedzams ar viegli izmantojamu Windows operētājsistēmu rīku, kas ļauj palaist programmas automātiski datora ieslēgšanas brīdī. Tas ir svarīgs process lai nodrošinātu ka klients vispirms savienosies ar administratīvo programmu pirms datoru var ilglaicīgi izmantot.

**Izvietošanas elementi:**

* Windows rīks “Startup applications” – Viegli ļauj pievienot programmas palaižamo failu mapē vai caur programmu “Task Manager”. Šis nodrošinās ka programma pati palaižas datora ieslēgšanā, tādējādi nodrošinot ka dators savienosies ar administratīvo programmu ieslēgšanas brīdī.

## Vienlaicīguma skatupunkta elementi

Programmas būtība ir dinamiskas situācijas kontrolēšana. Administratīvajai programmai ir jāspēj aktīvi uzņemt un uzturēt nenoteiktu skaitu savienoto klientu. Apskatot šos elementus no vienlaicīguma skatupunkta ir svarīgi, jo programmai jāspēj reaģētu uz jebkuru lietotāja darbību, jebkurā laikā, izpildot attiecīgo komandu un funkciju.

### Klientu skaits

Klientu skaitam nav ierobežojuma, tāpēc programmai ir dinamiski jāspēj pievienot jaunus klientu savienojumus, lietotājam netraucējot. Tas ir ātrs process un tas balstās uz klienta pašreizējo stāvokli.

**Vienlaicīguma elementi:**

* Automātiska klientu saņemšana – Programma pati saņem un noņem klientus no saskarnes saraksta.
* Klientu uzturēšana – Katra individuāla klienta stāvokļa vizualizācija un attēlošana, kā arī klienta savienojuma automātiskā noņemšana, ja pazūd savienojums.

### Vienlaicīgie procesi

Katram individuālajam klientam ir piešķirta sava laika vienība kurā tas spēj funkcionēt. Tādēļ ir nepieciešams noteikt, ka programma spēj vienlaicīgi uzturēt katra savienotā klienta laiku un stāvokli, netraucējot citiem aktīvajiem procesiem. Papildus tam, komandu izpilde var notikt ne tikai vienam datoram, bet arī vairākiem vienlaicīgi.

**Vienlaicīguma elementi:**

* Individuālā laika uzskaite – Spēja katram klientam noteikt savu laika periodu un nesajaukt aktīvos laika procesus un to izpildi.
* Vienlaicīga komandu izpilde vairākiem klientiem – Jānodrošina to, ka katra komanda tiek izpildīta uz katra no lietotāja atzīmētajiem klientiem.

### Sistēmas aktivitāte

Programmu ir paredzēts izmantot ilgstošos laika periodos, kuros ir iespējama maza aktivitāte. Tāpēc ir nepieciešams nodrošināt iespēju programmai saglabāt savu darbību ilgā laikā, bez funkcionalitātes pazušanas. Programmai ir jāuztur savi savienojumi kamēr tie ir aktīvi tīklā un jāļauj lietotājam aktīvi izmantot datoru nekoncentrējoties vai izmantojot programmu.

**Vienlaicīguma elementi:**

* Savienojumu saglabāšana – Process kurā savienojumi tiek aktīvi meklēti un atjaunoti, nebalstoties uz lietotāja pašreizējo aktivitāti.
* Procesu automatizācija – Ja lietotājs izvēlās izpildīt kādu komandu tad tā tiks izpildīta un stāvoklis atspoguļots saskarne, pēc kā lietotājs var aktīvi darīt paralēlas darbības.

# TimeStation funkcionālais projektējums

Administratīvā programma ir galvenais elements visā sistēmā. Tā nosaka kā strādā klienta programma. Funkciju sadalījums ir vienkāršs:

* Administratīvā programma – izveido serveri un uztur to; uztur klientu savienojumus; vizualizē klientus; izsūta lietotāja komandas; aprēķina naudas summu;
* Klienta programma – saņem komandas un izpilda tās; nosūta atlikušo laiku

Lai gan funkciju sadalījums ir vērsts uz vienu programmu, abas aktīvi strādā lai nodrošinātu darbību.

Abās programmas ir noteiktas funkcijas, kuras ir līdzīgas viena otrai, bet atbild viena nošķirta elementa pārvaldi. Tas pats arī attiecas uz savienojumu loģiku programmām, jo tas kā komandas tiek saņemtas un izsūtītas abām programmām ir identisks.

Lai labāk izprastu katru funkciju un kam tā ir nepieciešama, zemāk tiek aprakstītas visas programmas darbā iesaistītās funkcijas. Tās tiek aprakstītas ne tikai no pirmkoda skatupunkta, bet arī no nepieciešamības un tās iekšējās darbības.

Lai nebūtu jāpārkopē katras saistītās funkcijas pirmkodu, viss programmas un pašu funkcionālo funkciju pirmkods ir pieejams GitHub repozitorijā, kurš ir pieejams: <https://github.com/Rudis-P/Datoru_Kontrole>. Šeit var apskatīt visu programmas izstrādes ciklu un pašas funkcijas ja ir vajadzība tās apskatīt no programmas viedokļa.

## Servera izveide un savienojumu saņemšana

Galvenais programmas uzdevums ir sazināties ar klienta programmu. Tas tiek veikts izveidojot serveri uz galvenā datora, kurš tad spēj veikt laika kontroli.

### Servera izveide un savienojumu saņemšana būtība

Princips ir vienkāršs - kad tiek palaista programma, tā uzstāda uz noteiktā datora porta atvērtu tīkla savienojumu, kas var pieslēgties klienta programmas. Šis savienojums ir vienmēr ir atvērts un tas spēj izsūtīt komandas un saņemt laika atskaiti.

### Servera izveide un savienojumu saņemšana loģika

|  |  |  |
| --- | --- | --- |
|  | Blokshēma |  |
|  |  |  |
|  | Funkcijas ievades dati |  |
|  | Automātiskā funkcija, kurai nav ievade. |  |
|  | Funkcijas izvades dati |  |
|  | Saraksts ar klientiem. Ja ir jauns klients, tas tiek pievienots sarakstam un attēlots saskarnē ar tabulas elementu. |  |
|  | Saistītās funkcijas |  |
|  | Vārdu ielādēs, Programmas saskarnes ielāde, Laika atjaunošana. |  |
|  |  |  |

## Laika pievienošana un atjaunošana

Līdzko ir aktīvs savienojums, lietotājs var pievienot, atņemt vai labot laika pieslēgumu vienam, vai vairākiem klientiem.

### Servera izveide un savienojumu saņemšana būtība

Princips ir vienkāršs - kad tiek palaista programma, tā uzstāda uz noteiktā datora porta atvērtu tīkla savienojumu, kas var pieslēgties klienta programmas. Šis savienojums ir vienmēr ir atvērts un tas spēj izsūtīt komandas un saņemt laika atskaiti.

### Servera izveide un savienojumu saņemšana loģika

|  |  |  |
| --- | --- | --- |
|  | Blokshēma |  |
|  |  |  |
|  | Funkcijas ievades dati |  |
|  | Automātiskā funkcija, kurai nav ievade. |  |
|  | Funkcijas izvades dati |  |
|  | Saraksts ar klientiem. Ja ir jauns klients, tas tiek pievienots sarakstam un attēlots saskarnē ar tabulas elementu. |  |
|  | Saistītās funkcijas |  |
|  | Vārdu ielādēs, Programmas saskarnes ielāde, Laika atjaunošana. |  |
|  |  |  |

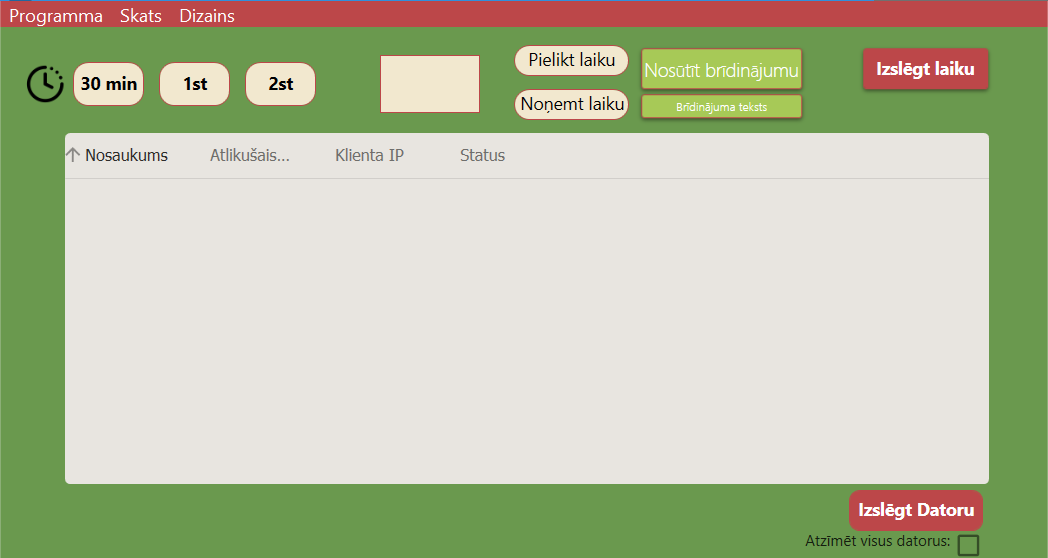
# Lietotāja saskarnes projektējums

Ņemot vērā ka programma tiek veidota iekš WPF, tad vizuālās saskarnes dizains tiks veidots iekš XAML valodas. Programmas lietotāja saskarnei ir vairāki elementi, kuri tiks aprakstīti šajā daļa, kopā ar attēlojumiem katram minētajam elementam, saniezot ieskatu ar konkrēto elementu saistītajām funkcijām.

## Lietotāja saskarne

Lietotāja saskarne sastāv no šādiem elementiem:

9

**

7

8

6

5

4

3

2

1

1. Fiksētās laika pogas, caur kurām var ātri pievienot norādīto laiku.
2. Logs laika skaitļa ievadei, kurā var norādīt minūtes pievienošanai vai atņemšanai.
3. Pogas blakus lauka darbības noteikšanai.
4. Brīdinājuma pogas, noklusējuma teksta vai lietotāja norādītam teksta nosūtīšanai.
5. Laika izslēgšana atzīmētajiem vienumiem.
6. Galvenais elements klientu attēlošanai, kurā savienotie klienti parādās kā rindas tabulā.
7. Datoru izslēgšanas poga.
8. Visu savienoto klientu ātrā atzīmēšana vienlaicīgai komandu izpildei.
9. Programmas izvēlnes rinda papildus funkciju un darbība izmantošanai.

## XAML struktūra

Programmas dizainam tiek izmantota XAML valoda ar noteiktām bibliotēkām. Šīs bibliotēkas sniedz papildus iespējas uzlabot un pilnveidot programmas ikdienas izmantošanu. Pamata elementi dizaina izveidei ir režģa struktūra, jeb <Grid>.

Papildus tam visā XAML dizaina failā tiek izmantota dinamiska pieeja noformējuma krāsām, kur katra elementa krāsai ir piešķirts nevis statisks krāsas kods, bet dinamisks mainīgais, kuru lietotājs spēj mainīt.

## Programmas krāsu dizain

Kā tika minēts iepriekšējā punktā, programmas dizainam ir iespējams noteikt jebkādu krāsu pēc izvēles. Šī izvēle atrodas zem programmas izvēles rindas elementa “Dizains”.

 Lietotājam ir iespējas nomainīt katru krāsas elementu, vai atiestatīt noklusējuma krāsas.

## Programmas redzamo elementu kontrole

Izmantojot izvēlnes rindu, ir iespējams noteikt atsevišķu klientu tabulas kolonu rādīšanu. Šī izvēle tiek saglabāta lietotāja iestatījumos.

 Lietotājam ir iespējas noteikt kuru kolonu rādīt vai paslēpt.

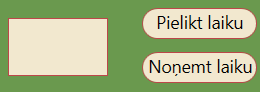
## Programmas darbības kontrole

Izmantojot izvēlnes rindu lietotājs var piekļūt programmas darbības iestatījumiem. Šie ļauj personalizēt programmas darbību.

 Lietotājam ir iespējas noteikt maksas skaitīšanas funkcijas aktivizēšanu un aizvēršanas apstiprināšanu. Papildus tam ir tūlītējā programmas aizvēršanas iespēja.

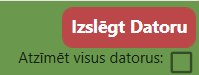
## Galvenās laika pogas

Lai atvieglotu lietotāja pieredzi, ir noteiktas trīs pogas, kuras sniedz ātru piekļuvi standarta laikiem. Šie laiki ir fiksēti un tos nevar mainīt. Laika pogām blakus ir laika ikona, kura attēlo pašreizēji izvēlētā klienta stāvokli, mainot savu aizpildījumu. Kur neizvēlētā situācijā ikonai ir caurspīdīga mugura, bet ja ir izvēlēts klients, tad mugura tiek iekrāsota attiecīgi pašreizējajam laika. Kā arī ir iespēja noteikt laika skaitli kuru var izmantot lai izmainīti atzīmētā klienta pašreizējo laiku.



## Klienta administrēšanas iespējas

Pēc vajadzības, ir iespējas nosūtīt brīdinājumu, vai pavisam izslēgt izvēlēto klientu. Papildus tam, ir iespēja izslēgt datoru pavisam un atzīmēt visus aktīvos savienojumus, ja ir vairāk par vienu.

## Klientu tabula

Galvenais elements programmai, kurš parāda aktīvos savienojumus tabulas veidā. Tabula spēj dinamiski pievienot un noņemt savienojumus un tai ir iespēja, ar labo peles taustiņu, atvērt logu kurā var ierakstīt noteiktā klienta identificējošu nosaukumu.

