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

**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](#_Toc157604178)

[1.1. Nolūks 1-1](#_Toc157604179)

[1.2. Darbības sfēra 1-1](#_Toc157604180)

[1.3. Definīcijas, akronīmi un saīsinājumi 1-1](#_Toc157604181)

[1.4. Saistība ar citiem dokumentiem 1-1](#_Toc157604182)

[1.5. Dokumenta raksturojums 1-1](#_Toc157604183)

[2. TimeStation vispārīgs apraksts 2-2](#_Toc157604184)

[2.1. TimeStation sadarbība ar citiem produktiem un sistēmām 2-2](#_Toc157604185)

[2.2. Projektēšanas ierobežojumi 2-2](#_Toc157604186)

[3. TimeStation arhitektūra 3-3](#_Toc157604187)

[4. TimeStation datu bāzes modelis 4-3](#_Toc157604188)

[5. TimeStation funkcionālais projektējums 5-3](#_Toc157604189)

[5.1. [Funkcijas nosaukums] 5-3](#_Toc157604190)

[5.1.1. [Funkcijas nosaukums] lietotāja saskarnes projektējums 5-3](#_Toc157604191)

[5.1.2. [Funkcijas nosaukums] algoritms 5-3](#_Toc157604192)

# 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, gan 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
* GitHub repozitorijs, kas satur versiju un izmaiņu vēsturi: <https://github.com/Rudis-P/Datoru_Kontrole>
* Changelog.docx

## 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.

## 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, izmantojamās tehnoloģijas un bibliotēkas.

### 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

*Sniedz shematisku attēlojumu un īsu aprakstu par izstrādājamās programmatūras arhitektūru.*

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

DIAGRAMMA UML??

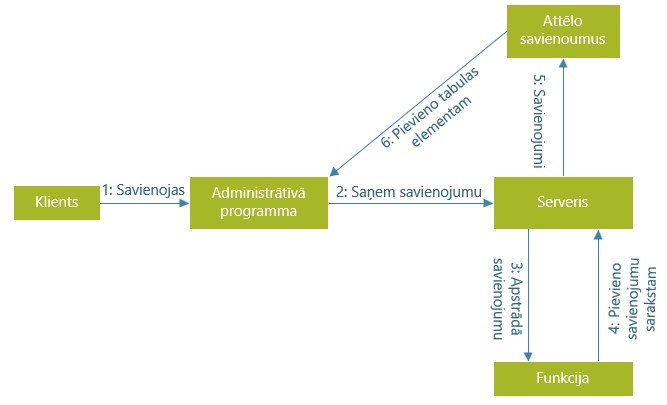
## Funkcionālā skatapunkta elementi

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 uskaitītas un aprakstītas, izsakot to būtību un vajadzību.

### Savienojuma loģika

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

Savienojums strādā no administratīvās programmas atvērtā datora porta, kuršs saņem savienojumus, pievienojot tos programmai. Attiecīgi caur so 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.

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

### Laika pievienošanas loģika

Līdz ko ir veiksmīgi izveiots 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.

**Funkcionālie elementi:**

* Laika pievienošana – No lietotāja puses tiek noteikta laika vienība, kas tiek nosūtīta klientam caur savienojumu.
* Laika noņemšana – Izmantojot to pašu pieeju kā pievienošanai, atlikušo laiku ir iespējam samazināt.
* Laika attēlošana – Vizuālajā vidē tiek attēlots atlikušais laiks, gan ar teksta rindu, gan ar krāsu palīdzību.
* Laika atjaunošana – Aktīvs reāllaika process, kuršs atjauno atlikušo laiku.

**Saistītās saskarnes elementi** –

# TimeStation sistēmas modeļi

# TimeStation funkcionālais projektējums

*Sniedz kopskatu par programmas kopējo struktūru – funkciju sadalījums/funkcionālie moduļi u.tml.*

## [Funkcijas nosaukums]

*Sniedz īsu konkrētās funkcijas vispārējo aprakstu.*

### [Funkcijas nosaukums] algoritms

|  |  |  |
| --- | --- | --- |
| Komanda – lietotājs nospiež attiecīgo pogu | Nosūta ziņu klienta programmai |  |
|  | Klients izpilda komandu |
| Laiks – izvēlētā laika vienība |  |

# [Funkcijas nosaukums] lietotāja saskarnes projektējums

*Apraksta konkrētās funkcijas* ***lietotāja saskarnes (interfeisa) projektējumu****, izveidojot saskarnes* ***vizuālo attēlojumu*** *un sniedzot tās aprakstu.*

## XAML

*Apraksta konkrētās funkcijas* ***izpildes scenārija/algoritma projektējumu****, izveidojot funkcijas izpildes scenārija/algoritma blokshēmu (-as) un sniedzot atbilstošu aprakstu.*

***Svarīgi!***

***Lietotāja saskarnes (5.1.1) un algoritma (5.1.2) projektējumos ir jāapraksta ievadāmo un izvadāmo datu saistību ar datubāzi (kas ir aprakstīta 4 nodaļā) (piemēru skat. failā PPA\_IAN\_darba)***