

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**UZŅĒMUMA “SAF TEHNIKA”
TERMOSKAPJU VADĪBAS SISTĒMA**

KVALIFIKĀCIJAS DARBS

Autors: **Druvis Cukurs**

Stud. apl. Nr.: dc13014

Darba vadītājs: Dr.dat. Dainis Dosbergs

RĪGA 2015

SATURA RĀDĪTĀJS

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	7
1.1. Ievads	7
1.1.1. Nolūks	7
1.1.2. Darbības sfēra	7
1.1.3. Definīcijas, akronīmi un saīsinājumi	7
1.1.4. Saistība ar citiem dokumentiem.....	9
1.1.5. Pārskats	9
1.2. Vispārējs apraksts.....	10
1.2.1. Produkta perspektīva.....	10
1.2.2. Produkta funkcijas	10
1.2.2. Lietotāju raksturiezīmes.....	13
1.2.3. Vispārējie ierobežojumi	13
1.3. Konkrētas prasības	13
1.3.1. Funkcionālās prasības	13
1.3.1.1. Saskarnes sadaļa “ <i>Status</i> ”	14
1.3.1.2. Saskarnes sadaļa “ <i>Configuration</i> ”	17
1.3.1.3. Saskarnes sadaļa “ <i>Modes</i> ”	21
1.3.1.4. Saskarnes sadaļa “ <i>Profiles</i> ”	26
1.3.1.5. Saskarnes sadaļa “ <i>History</i> ”	32
1.3.1.6. Saskarnes sadaļa “ <i>System health</i> ”	34
1.3.2. Nefunkcionālās prasības	36
1.3.2.1. Veiktspējas prasības	36
1.3.2.2. Pieejamība	36
1.3.2.3. Drošība.....	36
1.3.3. Ārējās saskarnes prasības.....	37
1.3.3.1. Lietotāja saskarne	37
1.3.4. Programmatūras saskarne	37
1.3.4.1. Sakaru saskarne	37

2.	PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	38
2.1.	Ievads	38
2.1.1.	Nolūks	38
2.1.1.	Darbības sfēra	38
2.1.2.	Definīcijas, akronīmi un saīsinājumi	38
2.2.	Saistība ar citiem dokumentiem	38
2.3.	Dekompozīcijas apraksts	39
2.3.1.	Moduļu dekompozīcija	39
2.3.1.1.	Saskarnes moduļi	40
2.3.1.2.	CGI moduļi	46
2.3.2.	Vienlaicīgo procesu dekompozīcija	54
2.3.2.1.	<i>main</i>	54
2.3.2.2.	<i>update_and_log</i>	54
2.3.2.3.	<i>process_profile</i>	54
2.3.3.	Datu dekompozīcija	58
2.3.3.1.	<i>Statusa informācija</i>	58
2.3.3.2.	<i>Iestatījumu informācija</i>	59
2.3.3.3.	<i>Interneta savienojuma iestatījumu informācija</i>	59
2.3.3.4.	<i>profiles.json</i>	60
2.3.3.5.	<i>temperature.csv</i>	62
2.3.3.6.	<i>system.csv</i>	63
2.4.	Trasējamības tabula	64
2.4.1.	Saskarnes sadaļa “ <i>Status</i> ”	64
2.4.2.	Saskarnes sadaļa “ <i>Configuration</i> ”	64
2.4.3.	Saskarnes sadaļa “ <i>Modes</i> ”	65
2.4.4.	Saskarnes sadaļa “ <i>Profiles</i> ”	65
2.4.5.	Saskarnes sadaļa “ <i>History</i> ”	66
2.4.6.	Saskarnes sadaļa “ <i>System health</i> ”	66

3.	TESTĒŠANAS DOKUMENTĀCIJA.....	67
3.1.	Ievads	67
3.2.	Testēšanas rezultāti	67
3.2.1.	“Status” un “Modes” sadaļas	67
3.2.2.	“Configuration” sadaļa	73
3.1.3.	“Profiles” sadaļa	76
3.1.4.	“History” sadaļa	78
3.1.5.	“System health” sadaļa	78
3.2.	Secinājumi	79
4.	PROJEKTA ORGANIZĀCIJA.....	80
5.	KVALITĀTES NODROŠINĀŠANA.....	81
5.1.	Dokumenta noformēšana.....	81
5.2.	Programmatūras koda noformēšana	81
5.2.1.	Mainīgo nosaukumu veidošana.....	81
5.2.2.	Komentāru rakstīšana.....	82
5.2.3.	Koda uzbūve	82
6.	KONFIGURĀCIJAS PĀRVALDĪBA	83
7.	DARBIETILPĪBAS NOVĒRTĒJUMS.....	84
8.	PROGRAMMATŪRAS KODA FRAGMENTI.....	87
9.	NOBEIGUMS	90

ANOTĀCIJA

Uzņēmums “SAF Tehnika” izstrādā dažādas iekārtas un rīkus platjoslu datu pārraidei un mobilo komunikāciju pieejamībai, garantējot to netraucētu darbību dažādās temperatūras amplitūdās. Lai pārliecinātos, ka iekārtas strādās noteiktajos apstākļos, tās no sākuma ir jānotestē, ko ir iespējams izdarīt ar izstrādāto sistēmu.

Izstrādātā sistēma ir iegultā sistēma, kas nozīmē, ka tā ir kādas lielākas sistēmas sastāvdaļa (šajā gadījumā to papildina izmantotie sensori temperatūras mērījumiem un termoskapji, kuros iekārtas tiek testētas) un ir paredzēta konkrētas darbības veikšanai – temperatūras kontrolēšanai termoskapī atbilstoši lietotāja izvēlētajam režīmam.

Sistēma tiek darbināta uz *Raspberry Pi* mikrodatora, un tās pamatfunktionalitāte ir programmēta C programmēšanas valodā. Priekš sistēmas ir izveidotas divas saskarnes – viena lietošanai no datora, bet otra lietošanai no 4 collu ekrāna, kas ir pievienots pašam mikrodatoram un atrodas blakus konkrētajam termoskapim, par kuru tas atbild.

Izstrādātā sistēma termoskapja apkalpotājam sniedz iespēju pārslēgt termoskapja režīmus, izveidot temperatūras profilus atkarībā no ierīcei paredzētās vides apstākļiem vai citiem faktoriem, kā arī sekot līdzi termoskapī notiekošajiem procesiem reālā laikā.

Atslēgvārdi: iegultā sistēma, Raspberry PI, Linux, JavaScript, FastCGI, C.

ABSTRACT

Company “SAF Tehnika” develops and manufactures devices for broadband data transmission and mobile communications, guaranteeing that they will function in different environment. To make sure that the devices will function as promised, they have to be tested. This can be done with the developed system.

The developed system is an embedded system which means that it is a part of a larger system (in this case the used sensors for the temperature measurements and the thermo chambers in which the devices are being tested make the complete system) and is with a dedicated function – to control the temperature within the thermo chamber according to the user’s selected mode.

The system operates on a *Raspberry Pi* microcomputer and it’s main functionality is written in the C programming language. Two graphical user interfaces have been developed – one for the use from a computer, the other one for the use from the 4 inch touchscreen mounted on the microcomputer which is located next to the thermo chamber it’s responsible for.

The developed system provides the ability for the thermo chamber attendant to switch the thermo chamber's modes, create temperature profiles depending on the device and in what sort of environment it will be used and monitor the thermo chamber's processes in real time.

Keywords: embedded system, Raspberry PI, Linux, JavaScript, FastCGI, C.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Izstrādātā programmatūras prasību specifikācija ir paredzēta uzņēmuma “SAF Tehnika” termoskapju vadības sistēmas programmatūras prasību definēšanai. Atbilstoši šai specifikācijai, kas ir izstrādāta kvalifikācijas darba autoram savstarpēji sadarbojoties ar uzņēmuma pārstāvjiem, tiks izstrādāta sistēma.

Šis dokuments ir domāts priekš programmatūras izstrādātāja un uzņēmuma pārstāvjiem, lai būtu precīzi noformulētas nepieciešamās prasības pret sistēmu.

1.1.2. Darbības sfēra

Uzņēmums “SAF Tehnika” nodrošina nepieciešamās iekārtas un rīkus platjoslu datu pārraidei un mobilo komunikāciju pieejamībai. Lai būtu pārliecība par klientiem garantēto iekārtas netraucētu darbību noteiktā temperatūras diapazonā, to ir nepieciešams iepriekš pārbaudīt, ko ir iespējams izdarīt ar termoskapju palīdzību.

1.1.3. Definīcijas, akronīmi un saīsinājumi

Iegultā sistēma - aparatūra un programmatūra, kas ir kādas lielākas sistēmas sastāvdaļa un kas paredzēta specifiska lietojuma funkcionēšanai, cilvēkam neiejaucoties

HTML (*Hypertext Markup Language*) - hiperteksta iezīmēšanas valoda, kas ir izstrādāta tīmekļa lapušu un citas pārlūkprogrammā attēlojamās informācijas glabāšanai.

CSS (*Cascading Style Sheets*) - kaskadētas stila lapas valoda, ko lieto, lai aprakstītu izskatu iezīmēšanas valodā veidotiem dokumentiem.

JavaScript – programmēšanas valoda, kas atļauj izveidot interaktīvas tīmekļa vietnes.

AJAX (*Asynchronous JavaScript and XML*) - tīmekļa vietņu izstrādes tehnoloģija, kas tiek izmantota *JavaScript* programmēšanas valodā.

Selenium – automatizēto testu izstrādes ietvars priekš tīmekļa lietojumprogrammām.

RPC (*Remote Procedure Call*) - attāls procedūras izsaukums, starpprocesu komunikācija, kas ļauj datorprogrammai izsaukt subrutīnu vai procedūru, lai to izpildītu citā adrešu telpā.

JSON (*JavaScript Object Notation*) - *JavaScript* programmēšanas valodā izmantotais datu strukturēšana formāts.

C – programmēšanas valoda.

CGI (*Common Gateway Interface*) - tīmekļa servera saskarne, kas ļauj ģenerēt tiešsaistes dokumentus, izmantojot izpildāmas programmas ārpus tīmekļa servera procesa.

FastCGI – protokols, kas nodrošina komunikāciju starp interaktīvām programmām un tīmekļa serveri.

Linux - operētājsistēma.

Pin – kontakttapiņa. Metāla adatiņa, no kādām veido ar pieslēgvietas kontaktligzdu saderīgo savienotāja spraudni.

GPIO (*General-Purpose InputOutput*) – mikroshēmas kontakttapiņa, kas var sūtīt vai saņemt signālus.

I2C (*I²C, Inter-Integrated Circuit*) – komunikācijas sistēma, kas pārsūta informāciju starp dažādām datora komponentēm.

HTTP (*HyperText Transfer Protocol*) – hiperteksta transporta protokols ir lietojumslāņa protokols, kas paredzēts datu apmaiņai starptīmekļa serveriem un pārlūkprogrammām.

Pavediens - thread. Laiksakritīgs process, kas ir kāda lielāka procesa vai programmas daļa. Šo terminu parasti izmanto vairākuzdevumu operētājsistēmu aprakstos, kad vairākas programmas daļas (pavedieni) var tikt izpildīti vienlaicīgi.

CSV (*Comma-separated Values*) – kompakts informācijas glabāšanas veids datnē tabulārā formā. Katrs ieraksts glabājas jaunā rindiņā, kur vērtības tiek atdalītas ar komatu vai kādu citu rakstu zīmi.

1.1.4. Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standarta LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis” prasības.

1.1.5. Pārskats

Dokuments sastāv no trīs nodaļām:

1.1. Ievads - aprakstīts programmaprodukts un definēta izmantotā terminoloģija.

1.2 Vispārējs apraksts - aprakstīta sistēmas darbība un uzskaitīta funkcionalitāte, kuru ir nepieciešams realizēt.

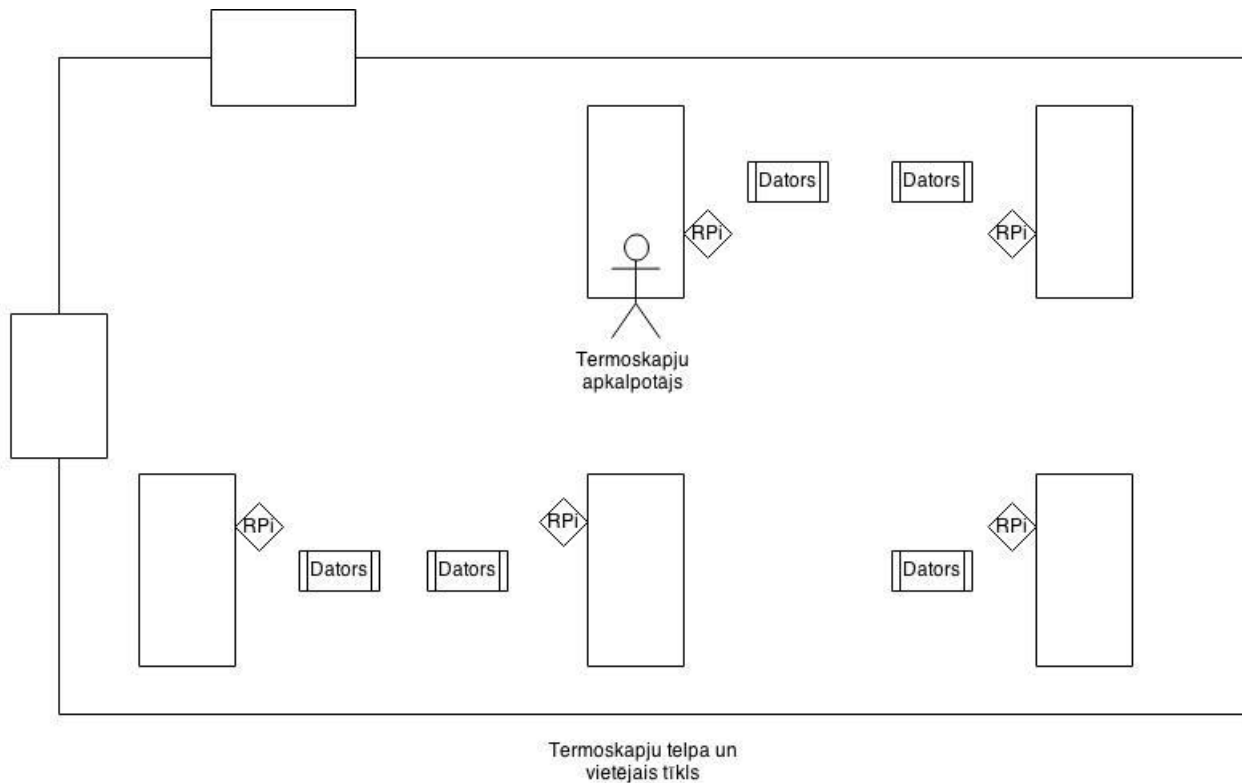
1.3. Konkrētās prasības - aprakstīta katra funkcija no sadaļas *1.2. Vispārējs apraksts* formā *Ievads, Ievade, Apstrāde, Izvaddati* un *Saistītās funkcijas*, kā arī nefunkcionālās prasības, kuras vispārējā aprakstā nav aprakstītas - veikspējas, pieejamības un drošības prasības.

1.2. Vispārējs apraksts

1.2.1. Produkta perspektīva

Šobrīd izstrādājamā programmatūra ir neatkarīga sistēma, kas nozīmē, ka šī sistēma nav saistīta ar kādām citām sistēmām. Izmantotais abstrakcijas līmenis to ļaus izmantot ar jebkuru citu sensoru ar I2C saskarni un jebkuru citu termoskapi ar GPIO kontrolējamām iekārtām.

Sistēma nodrošinās divas lietotāju saskarnes – vienu lietošanai no datora vietējā tīklā (vismaz pagaidām termoskapju telpas robežās), bet otru lietošanai no *Raspberry PI* pievienotā 4 collu LCD rezistīvā skārienjutīgā ekrāna blakus konkrētajam termoskapim, par kuru mikrodatore atbild. Ilustrācijas nolūkos seko grafika (*RPI* – *Raspberry Pi*).



1.2.2. Produkta funkcijas

Programmatūras produktam galvenokārt ir jānodrošina temperatūras kontrole termoskapī trīs režīmos – aizsardzības režīmā, stabilizācijas režīmā un temperatūras profila režīmā, kas ir jāveic ar sensoru mērījumu nolasīšanu termoskapī un termoskapju ierīču vadību.

Aizsardzības režīms ir noklusējuma režīms, kurā sistēma sāk savu darbu pēc tās ieslēgšanas, bet to ir iespējams aktivizēt jebkurā brīdī, pārtraucot stabilizācijas vai temperatūras profila režīma darbību. Šim režīmam atšķirībā no pārējiem diviem nav noteiktas mērķa

temperatūras, tā uzdevums nav uzturēt noteiktu temperatūru, bet gan vienkārši pasargāt termoskapī un tajā esošās ierīces no pārāk lielas sakaršanas.

Stabilizācijas režīmu ir iespējams uzskatīt par temperatūras profilu, kuram ir viens uzdevums ar noteiktu mērķa temperatūru, bet nenoteiktu uzdevumam veltīto laiku. Tā uzdevums ir noturēt termoskapī izvēlēto temperatūru līdz tiek ieslēgts aizsardzības vai profila režīms.

Termoskapis darbojas temperatūras profila režīmā, ja ir aktivizēts iepriekš lietotāja izveidots temperatūras profils. Temperatūras profils sastāv no viena vai vairākiem uzdevumiem, kam ir noteikta mērķa temperatūra un uzdevuma izpildei atvēlētais laiks.

Papildus temperatūras kontrolei, sistēmai ir jābūt konfigurējamai (cik bieži ir jāatjaunina statusa informācija, cik bieži ir jāveic mērījumi, lai koriģētu temperatūru, jābūt iespējai mainīt pieļaujamo temperatūras diapazonu pret mērķa temperatūru un temperatūras skalu, u.c. iestatījumus).

Ir jānodrošina funkcionalitāte temperatūras profilu izveidei, labošanai un dzēšanai. Lai, veidojot temperatūras profilu, būtu vieglāk uztvert tā izpildi, ir nepieciešams vizuāli attēlot temperatūras līkni grafikā.

Termoskapī notiekošie procesi un temperatūras svārstības ir jāattēlo lietotājam kā statusa informācija reālā laikā, tai skaitā temperatūras svārstības grafikā.

Lai būtu iespējams novērst radušās kļūdas vai arī nepieciešamības gadījumā uzlabot sistēmas darbību, ir nepieciešams informāciju par termoskapī notiekošajām darbībām, kā arī sistēmas stāvokļa informāciju saglabāt vēlākai analīzei notikumu vēstures žurnālā.

Sistēmai ir jānodrošina šāda funkcionalitāte:

Saskarnes sadaļā “*Status*”

- Statusa informācijas attēlošana
- Termoskapja temperatūras vēstures attēlošana grafikā
- Tekošās termoskapja temperatūras attēlošana grafikā
- Informācijas attēlošana par temperatūras punktu grafikā

Saskarnes sadaļā “*Configuration*”

- Iestatījumu informācijas attēlošana
- Iestatījumu rediģēšana

- Interneta savienojuma iestatījumu attēlošana
- Interneta savienojuma iestatījumu rediģēšana

Saskarnes sadaļā “*Modes*”

- Iespējamo režīmu un temperatūras profilu attēlošana
- Paziņojumu attēlošana izvēloties aizsardzības režīmu
- Aizsardzības režīma aktivizēšana
- Paziņojumu attēlošana izvēloties stabilizācijas režīmu
- Stabilizācijas režīma aktivizēšana
- Paziņojumu attēlošana izvēloties temperatūras profilu
- Temperatūras profila aktivizēšana

Saskarnes sadaļā “*Profiles*”

- Temperatūras profilu attēlošana
- Temperatūras profila uzdevumu attēlošana
- Temperatūras profila grafika attēlošana
- Informācijas attēlošana par temperatūras punktu grafikā
- Jauna temperatūras profila izveide
- Jauna temperatūras profila izveide, par pamatu izvēloties esošu profilu
- Temperatūra profila rediģēšana
- Temperatūras profila dzēšana

Saskarnes sadaļā “*History*”

- Notikumu vēstures attēlošana
- Notikumu vēstures šķirošana
- Notikumu vēstures datu lejupielāde

Saskarnes sadaļā “*System health*”

- Sistēmas stāvokļa informācijas attēlošana
- Sistēmas stāvokļa informācijas lejupielāde

1.2.2. Lietotāju raksturiezīmes

Izveidotā sistēma ir paredzēta uzņēmuma “SAF Tehnika” darbiniekiem, bet no lietotājiem netiek prasītas nekādas papildus zināšanas, netiek uzlikti specifiski ierobežojumi sistēmas darbības videi. Nepieciešamas pamatprasmes darbam ar datoru, uzstādīto interneta pārlūkprogrammu un interneta pieslēgums vietējam tīklam, lai piekļūtu termoskapja vadības sistēmai.

1.2.3. Vispārējie ierobežojumi

Termoskapju ierīču (dzesētāja, sildītāja un ventilatora) vadība notiek, izmantojot *GPIO* signālus, bet sensoru vadība norisinās, izmantojot *I2C* savienojuma saskarni.

Nefunkcionālo prasību un vispārējo ierobežojumu informācijai skatīt programmatūras prasību specifikācijas sadaļas “1.3. Konkrētas prasības” nodaļu “1.3.2. Nefunkcionālās prasības”.

1.3. Konkrētas prasības

1.3.1. Funkcionālās prasības

Pie funkciju ievaddatiem visi norādītie ievadlauki ir aizpildāmi obligāti, ja nav norādīts citādāk. Norādītie ievadlauku vērtību diapazoni ir ieskatot norādītās robežas (“no 1 līdz 10” nozīmē [1;10]).

Vienas un tās pašas funkcijas un to ievade, apstrāde, izvaddati attiecas uz abām saskarnēm – datora saskarni un Raspberry PI specializēto saskarni, ja nav norādīts citādāk.

Priekš Raspberry PI saskarnes sadaļas “*Info*” nav izveidota atsevišķa sadaļa šajā nodaļā, jo tā satur maz informāciju un to bija iespējams aprakstīt pie funkcijām, kas ar to ir saistītas.

1.3.1.1. Saskarnes sadaļa “Status”

1.3.1.1.1. Statusa informācijas attēlošana

Ievads
Funkcija nepieciešama, lai lietotājam attēlotu sistēmas un termoskapja statusa informāciju.
Ievade
-
Apstrāde
<p>Saskarnes sadaļai ielādējoties un ik pēc 5 sekundēm (pēc noklusējuma; intervālu ir iespējams mainīt iestatījumu sadaļā) sistēmai tiek veikts pieprasījums pēc statusa informācijas. Tā tiek apstrādāta un attēlota lietotājam:</p> <ul style="list-style-type: none">• Sistēmas darbības laiks kopš ieslēgšanas brīža nepieciešamības gadījumā tiek pārveidots līdz nedēļām.• Temperatūra pirms to attēlošanas tiek pārveidota pēc iestatījumos iestatītās temperatūras skalas un noapaļota ar precizitāti līdz desmitdaļai.
Izvaddati
<p>Iegūstot atbildi ar informāciju no servera, lietotājam tiek attēloti šādi lauki:</p> <p>Datora saskarnei:</p> <ul style="list-style-type: none">• <i>Device name</i> – termoskapja nosaukums;• <i>Uptime</i> – sistēmas darbības laiks kopš ieslēgšanas brīža;• <i>Sensors</i> – temperatūras sensoru nolasītie rādītāji;• <i>Temperature</i> – vidējā sensoru noteiktā temperatūra ledusskapī;• <i>Target temperature</i> – izvēlēta profila pašreizējā uzdevuma vai stabilizācijas režīma temperatūra (netiek attēlota aizsardzības režīmā);• <i>Mode</i> – aktīvais režīms,• <i>Profile</i> – profila nosaukums (netiek attēlots, ja nav ieslēgts profila režīms);• <i>Mode changed by</i> – IP adrese pēdējam lietotājam, kas nomainīja termoskapja režīmu (netiek attēlots tikko ieslēgtai sistēmai, kad aktīvais režīms vēl nav nomainīts);• <i>Heater</i> – sildītājs ieslēgts vai izslēgts (<i>On/Off</i>);• <i>Cooler</i> – dzesētājs ieslēgts vai izslēgts (<i>On/Off</i>);• <i>Fan</i> – ventilators ieslēgts vai izslēgts (<i>On/Off</i>). <p>Raspberry PI saskarnei:</p> <p>Saskarnes vidū tiek attēlots aktīvais termoskapja režīms vai temperatūras profils ar termoskapī esošo temperatūru. Zem tā ir attēlota ikona – liesma, ja termoskapī ierīces tiek karsētas, sniegpārsla, ja dzesētas.</p> <p>Ja ir ieslēgts temperatūras profils, saskarnes kreisajā apakšējā stūrī tiek attēlots temperatūras profila atkārtosšanās reižu skaits un cik reizes temperatūras profils vēl tiks atkārtots formātā “A/B”, kur A ir temperatūras profila atkārtosšanās reižu skaits, un B, cik reizes temperatūras profils tiks vēl atkārtots. Ja ir ieslēgts stabilizācijas režīms vai temperatūras profils, tad saskarnes labajā apakšējā stūrī tiek attēlota mērķa temperatūra.</p> <p>“Info” sadaļā tiek attēlots termoskapja nosaukums un sistēmas darbības laiks kopš ieslēgšanas brīža, nepieciešamības gadījumā pārveidots līdz nedēļām.</p>
Saistītās funkcijas
-

1.3.1.1.2. Termoskapja temperatūras vēstures attēlošana grafikā

Ievads
Funkcija nepieciešama, lai lietotājam būtu pārskatāma vizuāla ilustrācija par termoskapja temperatūras svārstībām.
Ievade
-
Apstrāde
Saskarnes “ <i>Status</i> ” sadaļai ielādējoties tiek veikts pieprasījums serverim pēc visiem esošajiem vēstures datiem, kas tiek apstrādāti un attēloti grafikā.
Izvaddati
Veiksmīgas funkcijas izpildes gadījumā: Datora saskarnē: Izveidots grafiks ar temperatūras līkni kopš sistēmas ieslēgšanas brīža, redzamībā attēlojot pēdējās 5 minūtes. Raspberry PI saskarnē: Izveidots grafiks ar temperatūras līkni par pēdējiem 150 ierakstiem. Neveiksmīgas funkcijas izpildes gadījumā: Vietā, kur vajadzēja tikt attēlotam temperatūras grafikam, ir attēlots paziņojums: “ <i>Loading graph. / Graph not available.</i> ”
Saistītās funkcijas
-

1.3.1.1.3. Tekošās termoskapja temperatūras attēlošana grafikā

Ievads
Funkcija nepieciešama, lai lietotājam būtu pārskatāma vizuāla ilustrācija par termoskapja temperatūras svārstībām. Grafika atjaunināšanu ir iespējams apturēt zem grafika nospiežot pogu “ <i>Live update is ON</i> ”. Kad tā ir nospiesta, grafika atjaunināšana ir atlikta līdz brīdim, kad tiek nospiesta poga “ <i>Live update is OFF</i> ”. No servera tiek iegūta informācija par iztrūkstošajiem līknes punktiem, tos attēlojot un turpinot atjaunināt grafiku.
Ievade
-
Apstrāde
No funkcijas 1.3.1.1.1. <i>Statusa informācijas attēlošana</i> iegūtajiem datiem tiek atlasīta un apstrādāta informācija par pievienojamo temperatūras punktu. Tiek pārbaudīts, vai temperatūra iekļaujas pieļaujamajā temperatūras diapazonā, ja nē, grafikā tiek attēlots sarkans temperatūras punkts.
Izvaddati
Grafikā attēlota tekošā temperatūra.
Saistītās funkcijas
1.3.1.1.1. <i>Statusa informācijas attēlošana</i>

1.3.1.1.4. Informācijas attēlošana par temperatūras punktu grafikā

Ievads
<p>Izvēloties konkrētu punktu līknē, zem grafika tiek attēlota papildus informācija.</p> <p>Svarīgākais paziņojuma veids ir brīdinājums par to, kurā brīdī temperatūra nav atradusies pieļaujamajā <i>deltas</i> diapazonā (kas grafikā tiek attēlots ar sarkanu punktu), un izrēķinātā nobīde no pieļaujamā diapazona.</p> <p>Raspberry PI saskarnē šī funkcija nav paredzēta.</p>
Ievade
<p>- (Izvēlēts punkts temperatūras profila grafikā.)</p>
Apstrāde
<p>Tiek iegūta informācija par temperatūras punktu, tā tiek apstrādāta un attēlota lietotājam:</p> <ul style="list-style-type: none">• Temperatūras punkta laiks tiek pārveidots formātā YYYY-MM-DD HH-MM-SS;• Tiek pārbaudīts, vai temperatūra iekļaujas pieļaujamajā temperatūras diapazonā, ja nē, tiek attēlots brīdinājums.
Izvaddati
<p>Lietotājam tiek attēloti šādi izvaddati:</p> <ul style="list-style-type: none">• datums un laiks,• aktīvais režīms vai temperatūras profils,• mērķa temperatūra (netiek attēlota aizsardzības režīmā),• temperatūra,• norādītas termoskapī ieslēgtās iekārtas (<i>Fan, Cooler, Heater</i>);• paziņojums vai brīdinājums (ja tāds konkrētajam punktam eksistē).
Saistītās funkcijas
<p>1.3.1.1.1. Statusa informācijas attēlošana</p>

1.3.1.2. Saskarnes sadaļa “*Configuration*”

Raspberry PI saskarnē šī sadaļa nav paredzēta.

1.3.1.2.1. Iestatījumu informācijas attēlošana

Ievads
Funkcija nepieciešama, lai lietotājam būtu iespējams attēlot sistēmas iestatījumus.
Ievade
-
Apstrāde
Iestatījumu informācija tiek apstrādāta un attēlota lietotājam: <ul style="list-style-type: none">• <i>Delta on</i> un <i>delta off</i> vērtības tiek pārveidotas lietotāja sistēmas iestatījumos iestatītajā temperatūras skalā.• Sistēmas laiks tiek formatēts formātā YYYY-MM-DD HH-MM-SS.
Izvaddati
Iegūstot atbildi ar informāciju no servera, lietotājam tiek attēloti šādi lauki: <ul style="list-style-type: none">• <i>Protocol version</i> – API versija (lietotājam nerediģējams lauks);• <i>Device name</i> – termoskapja nosaukums;• <i>Device serial number</i> – termoskapja sērijas numurs;• <i>Delta on</i> – temperatūras vērtība, kas ir pieļaujama zem mērķa temperatūras;• <i>Delta off</i> – temperatūras vērtība, kas ir pieļaujama virs mērķa temperatūras;• <i>Temperature unit</i> – temperatūras attēlošanas mērvienība;• <i>Temperature retrieve period</i> – temperatūras iegūšanas biežums no termoskapja (sekundēs);• <i>Status retrieve period</i> – statusa atjaunināšanas biežums saskarnē (sekundēs);• <i>System time</i> – sistēmā iestatītais laiks.
Saistītās funkcijas
-

1.3.1.2.2. Iestatījumu rediģēšana

Ievads
Funkcija nepieciešama, lai lietotājam būtu iespējams rediģēt sistēmas iestatījumus.
Ievade
Lietotājam ir iespējams rediģēt šādus sistēmas iestatījumus: <ul style="list-style-type: none">• <i>Device name</i>;• <i>Device serial number</i>;• <i>Delta on</i>;• <i>Delta off</i>;• <i>Temperature unit</i> (no izvēlnes);• <i>Temperature retrieve period</i>;• <i>Status retrieve period</i>;• <i>System time</i>.
Apstrāde
Mainītie iestatījumi tiek apstrādāti un nosūtīti serverim: <ul style="list-style-type: none">• “<i>Delta on</i>” un “<i>Delta off</i>” vērtības pirms nosūtīšanas serverim tiek pārvērstas Celsija skalā;• Ievadītais sistēmas laiks pirms nosūtīšanas serverim tiek pārvērsts par UNIX sekundēm. Serveris veic datu verifikāciju: <ul style="list-style-type: none">• vai ir ievadīts termoskapja nosaukums;• vai ir ievadīta <i>delta on</i> vērtība, un vai tas ir skaitlis robežā no 1 līdz 10;• vai ir ievadīta <i>delta off</i> vērtība, un vai tas ir skaitlis robežā no 1 līdz 10;• vai ir izvēlēta viena no pieejamajām temperatūras attēlošanas skalām – C (Celsija), F (Fārenheita) vai K (Kelvina);• vai ir ievadīts temperatūras iegūšanas intervāls, un vai tas ir skaitlis robežā no 1 līdz 60,• vai ir ievadīts statusa informācijas iegūšanas intervāls, un vai tas ir pozitīvs skaitlis;• vai ir ievadīts sistēmas laiks.
Izvaddati
Korektu ievaddatu gadījumā, lietotājam tiek attēlots paziņojums “ <i>The new configuration has been saved.</i> ”, un iestatījumu lauku vērtības tiek atjauninātas uz nomainītajām. Nekorektu ievaddatu gadījumā lietotājam tiek attēloti kļūdu paziņojumi, norādot uz konkrētiem laukiem, kas nav bijuši korekti aizpildīti un to iespējamajām vērtībām. Piemēram, “ <i>Field “delta on” doesn’t have a valid value. Possible values: [1;10].</i> ”
Saistītās funkcijas
-

1.3.1.2.3. Sistēmas laika lauka aizpilde ar lietotāja datora laiku

Ievads
Funkcija atvieglo sistēmas laika sinhronizēšanu ar lietotāja datora laiku, ko ir nepieciešams izdarīt pēc katras sistēmas ieslēgšanas reizes, jo tā atrodas vietējā tīklā, bet Raspberry Pi mikrodatoriem nav RTC (<i>Real Time Clock</i>) – pareizs laiks tiek iegūts no NTP (<i>Network Time Protocol</i>) servera.
Ievade
- (Lietotājs nospiež pogu “ <i>Set my time</i> ”.)
Apstrāde
Lietotāja datora laiks tiek pārveidots YYYY-MM-DD HH-MM-SS formātā.
Izvaddati
“ <i>System time</i> ” lauka vērtība tiek aizpildīta ar lietotāja datora laiku YYYY-MM-DD HH-MM-SS formātā.
Saistītās funkcijas
-

1.3.1.2.4. Interneta savienojuma iestatījumu attēlošana

Ievads
Funkcija ir nepieciešama, lai lietotājam attēlotu sistēmas interneta savienojuma iestatījumus.
Ievade
-
Apstrāde
Interneta savienojuma informācija tiek apstrādāta un attēlota lietotājam.
Izvaddati
Lietotājam tiek attēloti šādi lauki: Datora saskarnē: <ul style="list-style-type: none">• <i>Mode</i> (“<i>Static IP</i>” vai “<i>DHCP</i>”) - interneta savienojuma veids;• <i>Address</i> – sistēmas IP adrese;• <i>Netmask</i> – sistēmas IP adreses apakštīkls;• <i>Gateway</i> – sistēmas interneta savienojuma vārteja. Raspberry Pi saskarnē: “ <i>Info</i> ” sadaļā tiek attēlota sistēmas IP adrese.
Saistītās funkcijas
-

1.3.1.2.5. Interneta savienojuma iestatījumu rediģēšana

Ievads
Funkcija nepieciešama, lai lietotājam būtu iespējams rediģēt sistēmas interneta savienojuma iestatījumus.
Ievade
Lietotājam ir iespējams rediģēt šādus sistēmas interneta savienojuma iestatījumus: <ul style="list-style-type: none">• <i>Mode</i> (poga “<i>Static IP</i>” vai poga “<i>DHCP</i>”);• <i>Address</i>;• <i>Netmask</i>;• <i>Gateway</i>.
Apstrāde
Izvēloties nemainīgas IP adreses savienojuma veidu, serveris veic mainīto datu verifikāciju: <ul style="list-style-type: none">• vai ir izvēlēts nemainīgās IP adreses režīms;• vai ir ievadīta IP adrese un vai tā ir derīga IPv4 adrese;• vai ir ievadīts apakštīkls un vai tas ir derīgs apakštīkls;• vai ir ievadīta vārteja un vai tā ir derīga vārteja Izvēloties <i>DHCP</i> savienojuma veidu, tiek pārbaudīts, vai <i>DHCP</i> režīms jau nav ieslēgts. Ja tas ir ieslēgts, lietotājam tiek izvadīts paziņojums: “ <i>DHCP already active.</i> ”
Izvaddati
Veiksmīgu iestatījumu izmaiņu gadījumā, lietotājam tiek attēlots paziņojums “ <i>The network interface was updated successfully.</i> ”, un iestatījumu lauku vērtības tiek atjauninātas uz jaunajām vērtībām. Nekorektu ievaddatu gadījumā lietotājam tiek attēloti kļūdu paziņojumi, norādot uz konkrētiem laukiem, kas nav bijuši korekti aizpildīti. Piemēram, “ <i>Field "address" doesn't have a valid value. Must be a valid IPv4 address.</i> ”.
Saistītās funkcijas
-

1.3.1.3. Saskarnes sadaļa “Modes”

1.3.1.3.1. Iespējamo režīmu un temperatūras profilu attēlošana

Ievads
Funkcionalitāte nepieciešama, lai lietotājam attēlotu iespējamus režīmu un temperatūru profilus.
Ievade
-
Apstrāde
Piedāvātās 4 temperatūras stabilizācijas režīmam (55, 21, -5 un -33 Celsija skalā) tiek pārvērstas lietotāja iestatītajā temperatūras skalā, kā arī tiek pielāgota stabilizācijas temperatūras ievadlauka vērtību diapazons attiecīgajai temperatūras skalai. Izvēlnē saraksta veidā tiek attēloti esošie temperatūras profili.
Izvaddati
-
Saistītās funkcijas
-

1.3.1.3.2. Paziņojumu attēlošana izvēloties aizsardzības režīmu

Ievads
Funkcionalitāte nepieciešama, lai lietotāju informētu par darbības sekām, aktivizējot aizsardzības režīmu.
Ievade
Lietotājs no piedāvātajiem režīmiem izvēlas aizsardzības režīmu.
Apstrāde
-
Izvaddati
A. Ja aizsardzības režīms jau ir aktīvs, lietotājam tiek attēlots paziņojums “ <i>Protection mode already active.</i> ” un poga “ <i>Proceed</i> ”, lai aktivizētu izvēlēto režīmu, tiek attēlota kā izslēgta. B. Ja ir aktīvs stabilizācijas režīms, lietotājam tiek attēlots paziņojums: “ <i>About to activate protection mode. Stabilization at A B degrees will be stopped.</i> ” (kur A ir izvēlēta stabilizācijas temperatūra un B lietotāja iestatītā temperatūras skala). C. Ja ir aktīvs temperatūras profils, lietotājam tiek attēlots paziņojums: “ <i>About to activate protection mode. Profile "A" on cycle B from C will be stopped.</i> ” (kur A ir temperatūras profila nosaukums, B – cik reizes temperatūras profils ir atkārots, C – cik reizes temperatūras profils ir jāatkārto kopā.)
Saistītās funkcijas
1.3.1.3.3. Aizsardzības režīma aktivizēšana

1.3.1.3.3. Aizsardzības režīma aktivizēšana

Ievads
Funkcija nepieciešama, lai lietotājam būtu iespēja apturēt temperatūras profila vai stabilizācijas režīma darbību.
Ievade
- (Lietotājs nospiež pogu “ <i>Proceed</i> ”).
Apstrāde
Sistēmai tiek nosūtīts pieprasījums pāriet aizsardzības režīmā.
Izvaddati
Veiksmīga pieprasījuma gadījumā lietotājam tiek attēlots paziņojums: “ <i>Protection mode has been activated.</i> ”.
Saistītās funkcijas
-

1.3.1.3.4. Paziņojumu attēlošana izvēloties stabilizācijas režīmu

Ievads
Funkcionalitāte nepieciešama, lai lietotāju informētu par darbības sekām, aktivizējot stabilizācijas režīmu.
Ievade
Lietotājs no piedāvātajiem režīmiem izvēlas stabilizācijas režīmu, lietotājs izvēlas vienu no piedāvātajām stabilizācijas temperatūras iespējām vai arī lietotājs stabilizācijas temperatūras ievadlaukā ievada temperatūru.
Apstrāde
-
Izvaddati
Izvēloties stabilizācijas režīmu, lietotājam tiek attēlots paziņojums: “ <i>Either select or enter the stabilization temperature.</i> ” un poga “ <i>Proceed</i> ” nav iespējams nospiegt, kamēr nav izvēlēta stabilizācijas temperatūra.
Izvēloties stabilizācijas režīmu no izvēles iespējām vai arī ievadot temperatūru ievadlaukā, lietotājam tiek attēlots paziņojums: “ <i>About to activate stabilization mode at A B degrees.</i> ” (kur A ir stabilizācijas temperatūra, un B lietotāja izvēlēta temperatūras skala).
Ja stabilizācijas režīms jau ir aktīvs, lietotājam papildus vienam no augstākminētajiem paziņojumiem tiek attēlots paziņojums “ <i>Stabilization at A B degrees will be stopped.</i> ” (kur A ir stabilizācijas temperatūra, un B lietotāja izvēlēta temperatūras skala).
Ja ir aktīvs temperatūras profils, lietotājam tiek attēlots paziņojums: “ <i>Profile "A" on cycle B from C will be stopped.</i> ” (kur A ir temperatūras profila nosaukums, B – cik reizes temperatūras profils ir atkārts, C – cik reizes temperatūras profils ir jāatkārto kopā.).
Saistītās funkcijas
1.3.1.3.5. Stabilizācijas režīma aktivizēšana

1.3.1.3.5. Stabilizācijas režīma aktivizēšana

Ievads
Funkcija nepieciešama, lai lietotājam būtu iespēja ieslēgt stabilizācijas režīmu noteiktā temperatūrā.
Ievade
- (Lietotājs nospiež pogu “ <i>Proceed</i> ”.)
Apstrāde
Sistēmai tiek nosūtīts pieprasījums pāriet stabilizācijas režīmā ar norādīto temperatūru pārveidotu Celsija skalā.
Izvaddati
Veiksmīga pieprasījuma gadījumā lietotājam tiek attēlots paziņojums: “ <i>Stabilization at A B degrees has been activated.</i> ” (Kur A ir stabilizācijas temperatūra B temperatūras skala.)
Saistītās funkcijas
-

1.3.1.3.6. Paziņojumu attēlošana izvēloties temperatūras profilu

Ievads
Funkcionalitāte nepieciešama, lai lietotāju informētu par darbības sekām, aktivizējot temperatūras profilu.
Ievade
Lietotājs no piedāvātajiem režīmiem izvēlas profila režīmu vai arī lietotājs no temperatūras profilu izvēlnes izvēlas temperatūras profilu.
Apstrāde
-
Izvaddati
<p>Izvēloties profilu režīmu, lietotājam tiek attēlots paziņojums <i>“Select the profile to activate.”</i> un pogu <i>“Proceed”</i> nav iespējams nospiegt, kamēr nav izvēlēts temperatūras profils un temperatūras profila atkārtotās reižu skaits.</p> <p>Izvēloties temperatūras profilu no izvēlnes parādās ievadlauks, kur ievadīt profila atkārtotās reižu skaitu (0, lai profilu atkārtotu līdz režīma maiņai) un lietotājam tiek attēlots vai nu paziņojums</p> <ul style="list-style-type: none">• <i>“About to activate profile “A” to repeat B times.”</i> (kur A ir temperatūras profila nosaukums, B – cik reizes temperatūras profilu atkārtos), vai nu paziņojums• <i>“About to activate profile “A” to repeat till stopped.”</i> (kur A ir temperatūras profila nosaukums), ja temperatūras profila atkārtotās reižu skaita ievadlaukā ir ievadīta nulle. <p>Ja jau ir aktīvs temperatūras profils vai arī stabilizācijas režīmam, lietotājam papildus vienam no augstākminētajiem paziņojumiem tiek attēlots vai nu paziņojums</p> <ul style="list-style-type: none">• <i>“Profile “A” on cycle B from C will be stopped.”</i> (kur A ir temperatūras profila nosaukums, B – cik reizes temperatūras profils ir atkārtots, C – cik reizes temperatūras profils ir jāatkārto kopā.), vai nu paziņojums• <i>“Profile “A” will be stopped.”</i> (kur A ir temperatūras profils), ja temperatūras profils darbojas līdz režīma maiņai, vai nu paziņojums, vai nu paziņojums• <i>“Stabilization at A B degrees will be stopped.”</i> (kur A ir stabilizācijas temperatūra, un B lietotāja izvēlēta temperatūras skala).
Saistītās funkcijas
1.3.1.3.7. Temperatūras profila aktivizēšana

1.3.1.3.7. Temperatūras profila aktivizēšana

Ievads
Funkcija ir nepieciešama, lai lietotājs varētu termoskapī esošās ierīces pakļaut iepriekš izveidotam temperatūras profilam.
Ievade
- (Lietotājs nospiež pogu “ <i>Proceed</i> ”.)
Apstrāde
Tiek pārbaudīts vai: <ul style="list-style-type: none">• profils ar šādu nosaukumu eksistē;• temperatūras profila atkārtotās reižu skaits ir ne negatīvs skaitlis (nulle, lai atkārtotu līdz termoskapja režīma maiņai).
Izvaddati
Veiksmīga pieprasījuma gadījumā, lietotājam tiek attēlots paziņojums: “ <i>Profile “A” has been activated.</i> ” (kur A ir aktivizētā profila nosaukums.)
Saistītās funkcijas
-

1.3.1.4. Saskarnes sadaļa “Profiles”

Raspberry PI saskarnē šī sadaļa nav paredzēta.

1.3.1.4.1. Temperatūras profilu attēlošana

Ievads
Funkcija nepieciešama, lai lietotājam attēlotu esošos temperatūras profilus.
Ievade
-
Apstrāde
-
Izvaddati
Izvēlnē saraksta veidā tiek attēloti esošie temperatūras profili.
Saistītās funkcijas
-

1.3.1.4.2. Temperatūras profila uzdevumu attēlošana

Ievads
Funkcija nepieciešama, lai lietotājam attēlotu izvēlētā temperatūras profila uzdevumus.
Ievade
- (Izvēlēts temperatūras profils no izvēlnes.)
Apstrāde
Temperatūras profila uzdevumi tiek apstrādāti un attēloti. Uzdevumu mērķa temperatūra tiek attēlota lietotāja iestatījumos iestatītajā temperatūras skalā ar precizitāti viens ciparu aiz komata.
Izvaddati
Temperatūras profila uzdevumi tiek attēloti zem pieejamo profilu izvēlnes kā saraksts no 1 līdz N, kur N ir konkrētā temperatūras profila uzdevumu skaits.
Saistītās funkcijas
-

1.3.1.4.3. Temperatūras profila grafika attēlošana

Ievads
Funkcija nepieciešama, lai lietotājam būtu vieglāk uztvert temperatūras profila darbību, to attēlojot vizuāli kā līkni grafikā.
Ievade
Vai nu izvēlēts temperatūras profils no izvēlnes, vai arī temperatūras profila uzdevumi, veidojot vai labojot temperatūras profilu.
Apstrāde
Redzami temperatūras profila uzdevumi tiek apstrādāti un attēloti grafikā kā punkti līknē. Grafiks tiek atjaunots ar katru veikto izmaiņu uzdevumā vai arī ar katru pievienoto jauno uzdevuma temperatūras un uzdevumam paredzētā laika pāri.
Izvaddati
Izveidots temperatūras profila grafiks ar esošajiem temperatūras profila uzdevumiem.
Saistītās funkcijas
<i>1.3.1.4.4. Informācijas attēlošana par temperatūras punktu grafikā</i>

1.3.1.4.4. Informācijas attēlošana par temperatūras punktu grafikā

Ievads
Funkcija nepieciešama, lai lietotājam attēlotu informāciju par temperatūras punktu grafikā.
Ievade
- (Izvēlēts punkts temperatūras profila grafikā.)
Apstrāde
<p>Ja temperatūras profils ir izvēlēts no izvēlnes un nav labots, tad līknes attēlošanai tiek izmantota esošā informācija par temperatūras profilu, toties, ja tā uzdevumi ir mainīti vai arī profils tikai tiek veidots, tad līknes attēlošanai tiek izmantoti lietotāja ievadītie dati.</p> <p>Uzdevumu mērķa temperatūra tiek attēlota lietotāja iestatījumos iestatītajā temperatūras skalā ar precizitāti viens ciparu aiz komata.</p> <p><i>Date</i> lauka vērtība tiek attēlota YYYY-MM-DD HH-MM-SS formātā.</p>
Izvaddati
Lietotājam zem grafika tiek attēlots lauks “ <i>Temperature</i> ” ar konkrētā temperatūras profila uzdevuma temperatūru un lauks “ <i>Time</i> ” labošanas skatā ar vērtību, pēc cik minūtēm temperatūra tiktu sasniegta. Temperatūras profilu izvēles skatā lauka “ <i>Time</i> ” vietā tiek attēlots lauks “ <i>Date</i> ”, kas pretēji “ <i>Time</i> ” laukam parādītu relatīvo laiku – datumu un laiku, kad temperatūras profila uzdevuma temperatūra tiktu prognozēti sasniegta.
Saistītās funkcijas
-

1.3.1.4.5. Jauna temperatūras profila izveide

Ievads
Funkcija ir nepieciešama, lai lietotājs varētu izveidot jaunu temperatūras profilu.
Ievade
<ul style="list-style-type: none">• <i>Profile title</i> - profila nosaukums;• <i>Description</i> - īss apraksts, neobligāts ievades lauks;• N (kur N ir konkrēts skaitlis) skaits temperatūras profila uzdevumu ar temperatūras un uzdevumam paredzētā laika (minūtēs) pāriem.
Apstrāde
Tiek veikta ievadīto datu validācija: <ul style="list-style-type: none">• vai ir ievadīts temperatūras profila nosaukums, un vai temperatūras profils ar šādu nosaukumu jau neeksistē;• ir izveidots vismaz viens temperatūras profila uzdevums;• vai katra temperatūras profila uzdevuma temperatūra ir diapazonā no -55 grādiem līdz +50 grādiem Celsija skalā;• vai katra temperatūras profila uzdevuma atvēlētais laiks ir pozitīvs skaitlis līdz 2000.
Izvaddati
Korektu ievaddatu gadījumā, lietotājam tiek attēlots paziņojums: <i>“Profile "A" has been added.”</i> (kur A ir temperatūras profila nosaukums). Nekorektu ievaddatu gadījumā lietotājam tiek attēlots paziņojums, cik ir bijušas ievaddatu kļūdas, kā arī pie katra uzdevuma, kura ievadlauki nav saturējuši derīgas vērtības, tiek attēlots attiecīgais kļūdas paziņojums, norādot iespējamo vērtību diapazonu. Iespējamie kļūdas paziņojumi: <ul style="list-style-type: none">• <i>“Invalid temperature. Possible values [-50.0;55.0] (C), [-58.0;131.0] (F), [223.1;328.1] (K).”</i>• <i>“Invalid time. Possible values [1;2000].”</i>
Saistītās funkcijas
-

1.3.1.4.6. Jauna temperatūras profila izveide, par pamatu izvēloties esošu profilu

Ievads
Funkcionalitāte nepieciešama, lai lietotājam būtu ērtāk izveidot jaunu temperatūras profilu, par pamatu izvēloties jau esošu profilu.
Ievade
Sākotnēji tiek izvēlēts temperatūras profils no izvēlnes un atvērts labošanas režīmā, nospiežot pogu <i>“Edit profile”</i> . Labošanas režīmā poga <i>“Add profile”</i> tiek pārsaukta par <i>“Use as a template”</i> . To noklikšķinot, <i>“Save profile”</i> poga vairs nenodrošinās izvēlēta profila rediģēšanu, bet gan jauna profila izveidi ar esošajiem datiem. Tālākie ievaddati sakrīt ar funkcijas <i>“1.3.1.3.7. Jaunu temperatūras profila izveide”</i> ievaddatiem.
Apstrāde
Funkcijas apstrāde sakrīt ar funkcijas <i>“1.3.1.4.5. Jauna temperatūras profila izveide”</i> apstrādi.
Izvaddati
Korektu ievaddatu gadījumā, lietotājam tiek attēlots paziņojums: <i>“Profile “A” has been added.”</i> (kur A ir temperatūras profila nosaukums). Nekorektu ievaddatu gadījumā, izvaddati sakrīt ar funkcijas <i>“1.3.1.4.5. Jauna temperatūras profila izveide”</i> izvaddatiem.
Saistītās funkcijas
-

1.3.1.4.7. Temperatūras profila rediģēšana

Ievads
Funkcija nepieciešama, lai lietotājs varētu rediģēt jau esošu temperatūras profilu.
Ievade
- (Izvēlēts temperatūras profils no izvēlnes un atvērts labošanas režīmā.)
Apstrāde
Funkcijas apstrāde sakrīt ar funkcijas <i>“1.3.1.4.5. Jauna temperatūras profila izveide”</i> apstrādi, tikai tiek pārbaudīts vai norādītais temperatūras profils eksistē.
Izvaddati
Korektu ievaddatu gadījumā, lietotājam tiek attēlots paziņojums: <i>“Profile “A” has been successfully edited.”</i> (kur A ir temperatūras profila nosaukums). Nekorektu ievaddatu gadījumā, izvaddati sakrīt ar funkcijas <i>“1.3.1.4.5. Jauna temperatūras profila izveide”</i> izvaddatiem.
Saistītās funkcijas
-

1.3.1.4.8. Temperatūras profilu dzēšana

Ievads
Funkcija nepieciešama, lai lietotājam būtu iespējams izdzēst temperatūras profilu.
Ievade
- (Izvēlēts temperatūras profils no izvēlnes.)
Apstrāde
Tiek pārbaudīts vai eksistē temperatūras profils ar šādu nosaukumu.
Izvaddati
Veiksmīga pieprasījuma gadījumā, lietotājam tiek attēlots paziņojums: <i>“Profile “A” was has been removed.”</i> (kur A ir profila nosaukums, kas tika izdzēsts.)
Saistītās funkcijas
-

1.3.1.5. Saskarnes sadaļa “History”

Raspberry PI saskarnē šī sadaļa nav paredzēta.

1.3.1.5.1. Notikumu vēstures attēlošana

Ievads
Funkcija nepieciešama, lai lietotājam attēlotu notikumu vēstures informāciju.
Ievade
Notikumu vēstures attēlošanas lapaspuse (atkarībā no atzīmēto ierakstu skaita vienā lapā (pēc noklusējuma tiek attēloti 20 ieraksti lapaspusē), sākotnēji tiek attēlota 1. lapaspuse).
Apstrāde
Tiek apstrādāta notikumu vēstures informācija un attēlota sekojošās kolonās: <ul style="list-style-type: none">• <i>Nr</i> – notikumu vēstures ieraksta numurs;• <i>Uptime</i> – laiks sekundēs kopš sistēmas ieslēgšanas brīža, nepieciešamības gadījumā pārveidots līdz nedēļām;• <i>Local time</i> – sistēmas datums un laiks <i>YYYY-MM-DD HH-MM-SS</i> formātā;• <i>Logging level</i> – notikumu vēstures ieraksta līmenis līmenis (iespējamie ierakstu līmeņi:<ul style="list-style-type: none">• <i>Error</i> – kļūda;• <i>Warning</i> – brīdinājums;• <i>Update</i> – atjauninājums (iestatījumu vai temperatūras profila);• <i>Info</i> – informācija (ieslēgts režīms, pabeigts temperatūras profila uzdevums);• <i>Debug</i> – statusa informācija.);• <i>Source</i> – ieraksta vai darbības veicējs (iestatījumu vai režīma maiņas gadījumā tā ir lietotāja IP adrese, citādāk temperatūras profila nosaukums, vārds “<i>System</i>” vai tukšums);• <i>Status</i> – statusa kolona satur informāciju par esošo temperatūru “<i>T</i>”, mērķa temperatūru “<i>TT</i>” (abas temperatūras ir attēlotas iestatījumos norādītajā temperatūras skalā) un par to, kuras iekārtas termoskapī tajā brīdī ir bijušas ieslēgtas – “<i>H</i>” (sildītājs), “<i>C</i>” (dzesētājs), “<i>F</i>” (ventilators).• <i>Message</i> – iespējams paziņojums.
Izvaddati
Tiek attēloti pirmie 20 ieraksti.
Saistītās funkcijas
1.3.1.5.2. Notikumu vēstures šķirošana

1.3.1.5.2. Notikumu vēstures šķirošana

Ievads
Funkcija nepieciešama, lai lietots varētu atlasīt tikai vēlamu informāciju no notikumu vēstures datiem.
Ievade
Atzīmētās notikumu vēstures datu kategorijas.
Apstrāde
No jau apstrādātajiem vēstures notikumu datiem tiek atlasīti tie ieraksti, kas atbilst lietotāja atzīmētajām kategorijām.
Izvaddati
Tiek attēloti A (kur A ir lietotāja atzīmēto ierakstu skaits vienā lapā - 10, 20 vai 50) pirmie ieraksti.
Saistītās funkcijas
1.3.1.5.1. Notikumu vēstures attēlošana

1.3.1.5.3. Notikumu vēstures lejupielāde

Ievads
Funkcija nepieciešama, lai būtu iespējams lejupielādēt visu notikumu vēstures informāciju vienkopus uz lietotāja datora, pārveidotu no CSV formāta formātā: <i>Notikuma vēstures ieraksta numurs – Sistēmas laiks, kad ir veikts ieraksts – Ieraksta līmenis – Ieraksta avots - T: temperatūra TT: mērķa temperatūra (ja ieraksta avots nav aizsardzības režīms) - H C F – ziņojums (ja tāds ir).</i>
Ievade
- (Lietotājs nospiež pogu.)
Apstrāde
Notikumu vēstures dati tiek saformatēti norādītajā formātā, lejupielādējamā datne tiek nosaukta ar šī brīža laiku formātā YYYY-MM-DD HH-MM-SS.
Izvaddati
-
Saistītās funkcijas
-

1.3.1.6. Saskarnes sadaļa “System health”

Raspberry PI saskarnē šī sadaļa nav paredzēta.

1.3.1.6.1. Sistēmas stāvokļa informācijas attēlošana

Ievads
Funkcija nepieciešama, lai lietotājam būtu iespējams novērtēt sistēmas stāvokli.
Ievade
<ul style="list-style-type: none">Sistēmas stāvokļa attēlošanas lapaspuse (atkarībā no atzīmēto ierakstu skaita vienā lapā (pēc noklusējuma tiek attēloti 20 ieraksti lapaspusē); sākotnēji tiek attēlota 1. lapaspuse).
Apstrāde
Tiek apstrādāta sistēmas stāvokļa informācija un attēlota sekojošās kolonās: <ul style="list-style-type: none"><i>Nr</i> – sistēmas stāvokļa ieraksta numurs;<i>Uptime</i> – laiks sekundēs kopš sistēmas ieslēgšanas brīža, nepieciešamības gadījumā pārveidots līdz nedēļām;<i>Local time</i> – sistēmas datums un laiks <i>YYYY-MM-DD HH-MM-SS</i> formātā;<i>Daemon RAM (KB)</i> – sistēmas patērētā operatīvā atmiņa kilobaitos.<i>Free RAM (KB)</i> – serverim pieejamā operatīvā atmiņa kilobaitos.<i>Daemon CPU (%)</i> – sistēmai patērētie procesora resursi no kopējiem resursiem (procentuāli).<i>Total CPU (%)</i> – kopējā servera procesora noslodze (procentuāli).<i>Processes</i> – serverī notiekošo procesu skaits.
Izvaddati
Tiek attēloti pirmie 20 ieraksti.
Saistītās funkcijas
-

1.3.1.6.2. Sistēmas stāvokļa informācijas lejupielāde

Ievads
Funkcija nepieciešama, lai būtu iespējams lejupielādēt visu sistēmas stāvokļa informāciju vienkopus uz lietotāja datora, pārveidotu no CSV formāta formātā: <i>Sistēmas stāvokļa informācijas ieraksta numurs – Sistēmas laiks, kad ir veikts ieraksts – Process RAM: sistēmas patērētā operatīvā atmiņa kilobaitos KB – Free RAM: serverim pieejamā operatīvā atmiņa kilobaitos KB – Daemon CPU: serverim patērētie procesora resursi no kopējiem resursiem (procentuāli) % - Total CPU: kopējā servera procesora noslodze (procentuāli) % - Processes: serverī notiekošo procesu skaits.</i>
Ievade
- (Lietotājs nospiež pogu.)
Apstrāde
Notikumu vēstures dati tiek saformatēti norādītajā formātā, lejupielādējamā datne tiek nosaukta ar šī brīža laiku formātā YYYY-MM-DD HH-MM-SS.
Izvaddati
-
Saistītās funkcijas
-

1.3.2. Nefunkcionālās prasības

1.3.2.1. Veiktspējas prasības

Sistēmas darbība tiek nodrošināta ar Raspberry PI B+ mikrodatoru, kuram ir 700 MHz single-core ARM1176JZF-S procesors (kas līdzinās 300 MHz II Pentium procesoram), 512 MB operatīvā atmiņa, kas reizē arī nodrošina grafisko apstrādi, kas nozīmē to, ka atmiņas un skaitļošanas resursi ir nelieli un pēc iespējas vairāk ir jāmēģina atlikt skaitļošanas procesu uz lietotāja datora, ne pašam serverim (visa informācijas apstrāde attēlošanai, notikumu vēstures sadalīšana lapās), lai nodrošinātu pilnvērtīgu funkcionalitāti, kā arī pozitīvu lietotāju pieredzi.

Serverim ir jāveic visas operācijas (termoskapja temperatūras kontrolēšana, notikumu vēstures uzturēšana, statusa informācijas sagatavošana un atbildes sagatavošana pieprasījumiem) 1 sekundes laikā, kas ir paredzēts biežākais statusa informācijas iegūšanas biežums lietotājam, lai attēlotu termoskapja temperatūru grafikā.

Maksimālās slodzes apstākļi:

- temperatūras iegūšana ik pēc 1 sekundes un
- statusa informācijas iegūšana ik pēc 1 sekundes vienlaicīgi 5 lietotājiem.

1.3.2.2. Pieejamība

Tā kā sistēma tiek izmantota produktu ražošanā no tās tiek sagaidīta netraucēta darbība cauru diennakti vismaz divus mēnešus. Maksimāli pieļaujama sistēmas pamatfunkcionalitātes atjaunošanas laiks ir 1 diena, bet nebūtiskas problēmas, kas neietekmē sistēmas pamatfunkcionalitāti, ir jāatrisina nedēļas laikā.

1.3.2.3. Drošība

1.3.2.3.1. Piekļuve

Piekļuve sistēmai ir paredzēta tikai vietējā tīklā, kas praktiski izslēdz iespēju no ārējās vides draudiem.

1.3.2.3.2. Dati

Tā kā piekļuve sistēmai ir iespējama tikai vietējā tīklā, tas nozīmē, ka sistēmas drošība ir jānodrošina vairāk pret netīšiem nekā tīšiem pāra darījumiem (piemēram, nekorektu ievaddatu

gadījumā, vai arī, ja ne ar nolūku ir sabojāta *JSON* struktūra iestatījumu vai profilu datnē pašrocīgi to rediģējot), tādēļ sistēmai ir jāpārbauda visi lietotāja ievadītie dati, kā arī, ja sistēmas darbības laikā tiek izmantota informācija, kas var tikt pašrocīgi rediģēta, tā ir jāpārbauda atkārtoti. Drošības risinājumi ir jārealizē tā, ka sistēmas darbība netiek pārtraukta nevienā brīdī, kā arī par šādiem kļūdu gadījumiem tiek veikti saprotami ieraksti notikumu vēstures datos.

1.3.3. Ārējās saskarnes prasības

1.3.3.1. Lietotāja saskarne

Ir nepieciešams izveidot divas lietotāja saskarnes – vienu, lietojot no datora, otru vienkāršotu un pielāgotu *Raspberry PI 4* collu ekrānam ar ne tik lielu funkcionalitāti – bez iestatījumu un žurnālu sadaļām. Saskarnēm ir jābūt vienkāršām un viegli saprotamām.

1.3.4. Programmatūras saskarne

1.3.4.1. Sakaru saskarne

Komunikācija ar sistēmu ir jānodrošina, izmantojot *HTTP* protokolu un *JSON* datu strukturēšanas formātu.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Programmatūras projektējuma apraksta, jeb PPA, izveides nolūks ir tehniski konkretizēt programmatūras prasību specifikācijā (PPS) esošo informāciju, atvieglojot sistēmas izstrādes procesu.

2.1.1. Darbības sfēra

Skatīt programmatūras prasību specifikācijas nodaļas “1.1. Ievads” sadaļu “1.1.2. Darbības sfēra”.

2.1.2. Definīcijas, akronīmi un saīsinājumi

Skatīt programmatūras prasību specifikācijas nodaļas “1.1. Ievads” sadaļu “1.1.3. Definīcijas, akronīmi un saīsinājumi”.

2.2. Saistība ar citiem dokumentiem

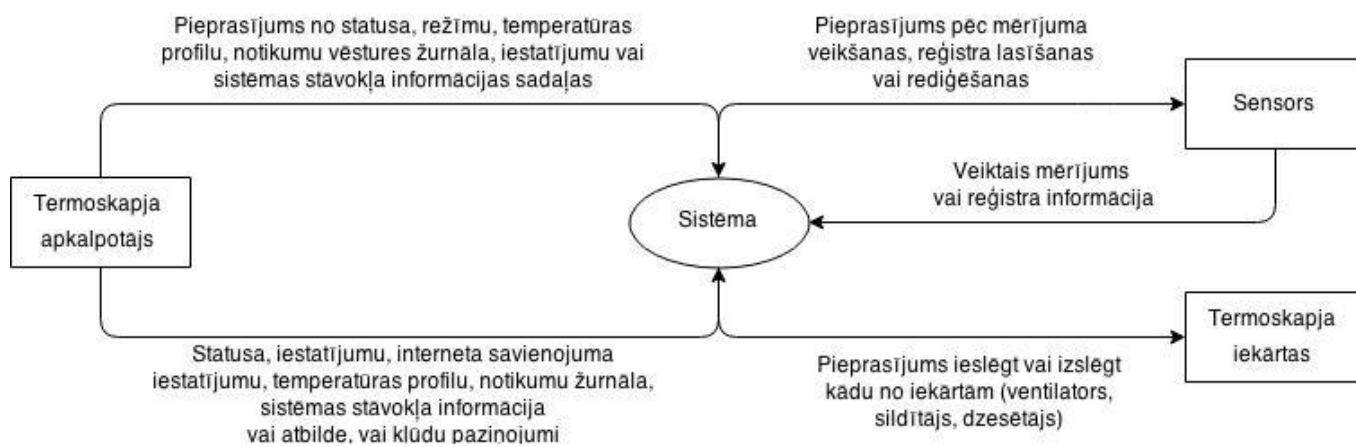
Dokumenta noformēšanā ievērotas standarta 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai” prasības.

2.3. Dekompozīcijas apraksts

2.3.1. Moduļu dekompozīcija

Moduļu pārskatāmības un ērtības nolūkos tie ir sadalīti divās sadaļās – saskarnei izmantotajos moduļos, kas atbild par pieprasījuma veikšanu sistēmas CGI modulim un informācijas attēlošanu un CGI moduļos, kas apstrādā saņemto informāciju un pieprasījumu, sniedzot uz to atbildi. Saskaņģu moduļu datu plūsmu diagrammas apzināti nav attēlotas detalizētāk, lai neapgrūtinātu sistēmas shēmas uztveršanu.

Zemāk ir attēlots sistēmas 0. līmeņa datu plūsmas diagramma. 1. līmeņa datu plūsmas diagramma ir attēlota moduļu dekompozīcijas beigās, kas reizē attēlo arī *main* procesu (sistēmas vienlaicīgie procesi ir aprakstīti un attēloti nākamajā sadaļā “2.3.2. Vienlaicīgo procesu dekompozīcija”).



Sistēmas 0. līmeņa DPD

2.3.1.1. Saskarnes moduļi

2.3.1.1.1. *status.inc*

Modulis ir paredzēts, lai lietotājam attēlotu termoskapja un sistēmas statusa informāciju, ieslēgto režīmu un tā stāvokli, šo informāciju pieprasot no servera ik pēc noteikta laika perioda, ko lietotājs var iestatīt iestatījumu sadaļā.

Blakus vai arī zem statusa informācijas (atkarībā no lietotāja monitora izšķirtspējas) ir attēlots termoskapja temperatūras grafiks, ar sarkaniem punktiem līknē attēlojot brīdinājumus vai kļūdas situācijas.

Par temperatūras punktu grafikā ir iespējams iegūt informāciju, uz tā uzbraucot ar kursoru.

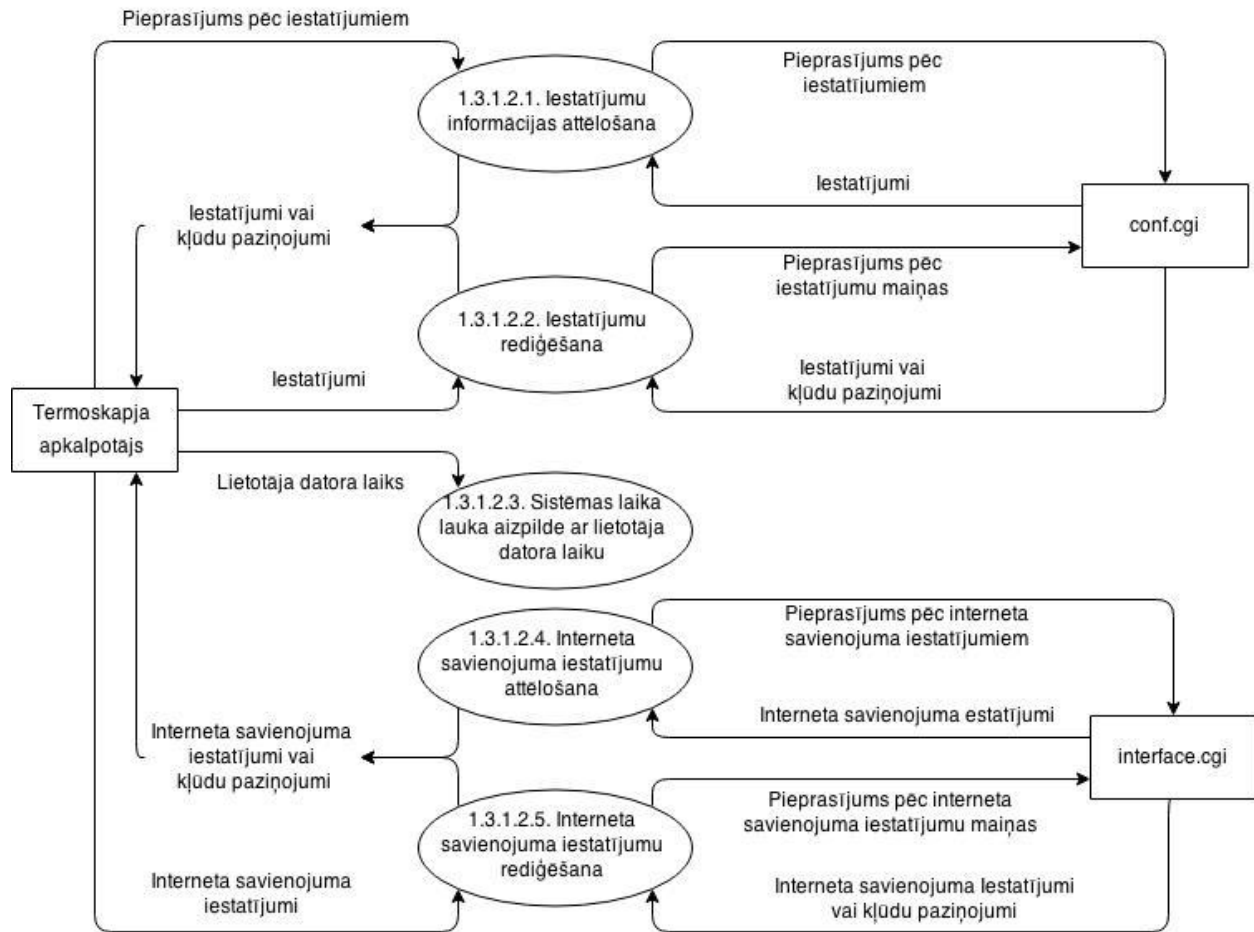


'Status' sadaļas 1. līmeņa DPD

2.3.1.1.2. *conf.inc*

Modulis ir paredzēts, lai lietotājam attēlotu sistēmas un interneta savienojuma iestatījumus un ļautu tos rediģēt.

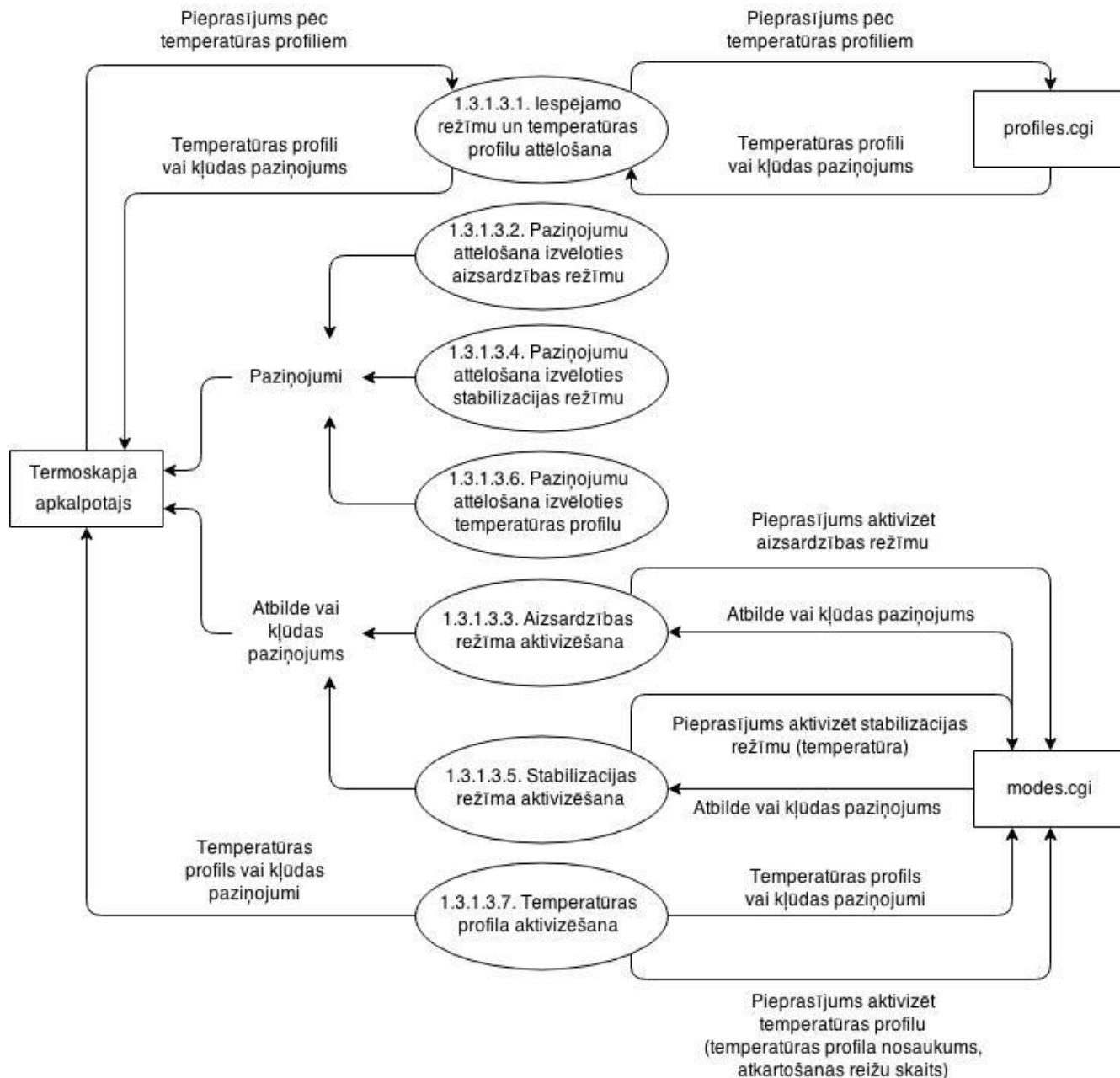
Iestatījumos ir iespējams nomainīt biežumu, cik bieži iegūt statusa informāciju un cik bieži veikt temperatūras mērījumus termoskapī, sistēmā attēloto temperatūras skalu, pieļaujamo temperatūras diapazonu, kā arī sistēmas laiku un interneta savienojumu iestatījumus.



'Configuration' sadaļas 1. līmeņa DPD

2.3.1.1.3. *modes.inc*

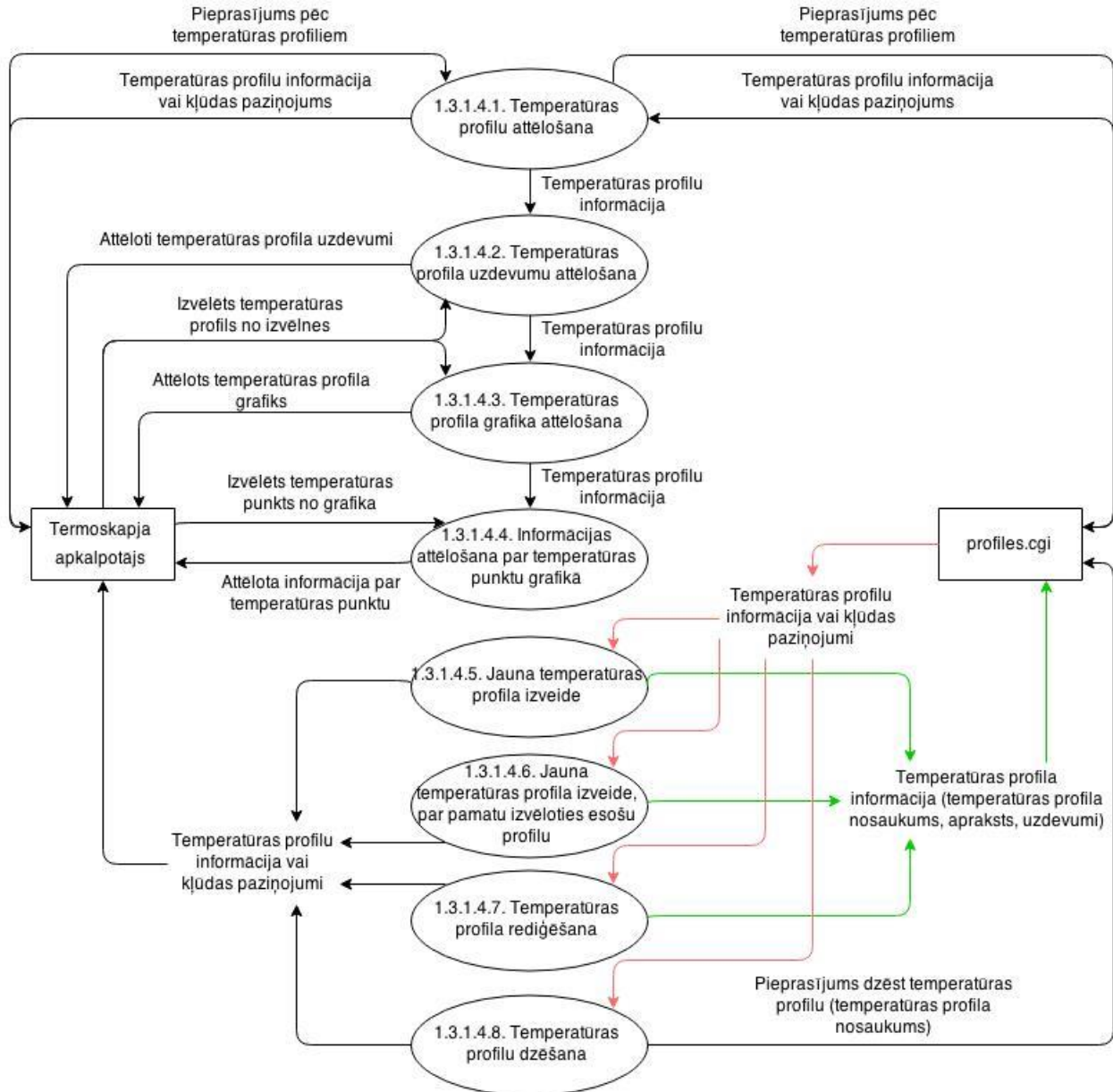
Modulis ir paredzēts, lai lietotājam attēlotu iespējamās termoskapja režīmus un temperatūras profili, ļautu tos mainīt, ieslēdzot aizsardzības, stabilizācijas vai profila režīmu, un attēlotu attiecīgus paziņojumus par lietotāja darbības sekām, piemēram, paziņojot, ka tiks pārtraukts aktīvais temperatūras profils, ja tāds ir ieslēgts.



'Modes' sadaļas 1. līmeņa DPD

2.3.1.1.4. *profiles.inc*

Modulis ir paredzēts, lai lietotājam attēlotu izveidotos temperatūras profilus, ļautu tos rediģēt, dzēst vai izveidot jaunus. Katram temperatūras profilam blakus tiek attēlots tā grafiks, kā arī veidojot vai labojot temperatūras profilu, grafiks tiek atjaunināts ar katru pievienoto vai laboto temperatūras profila uzdevumu.

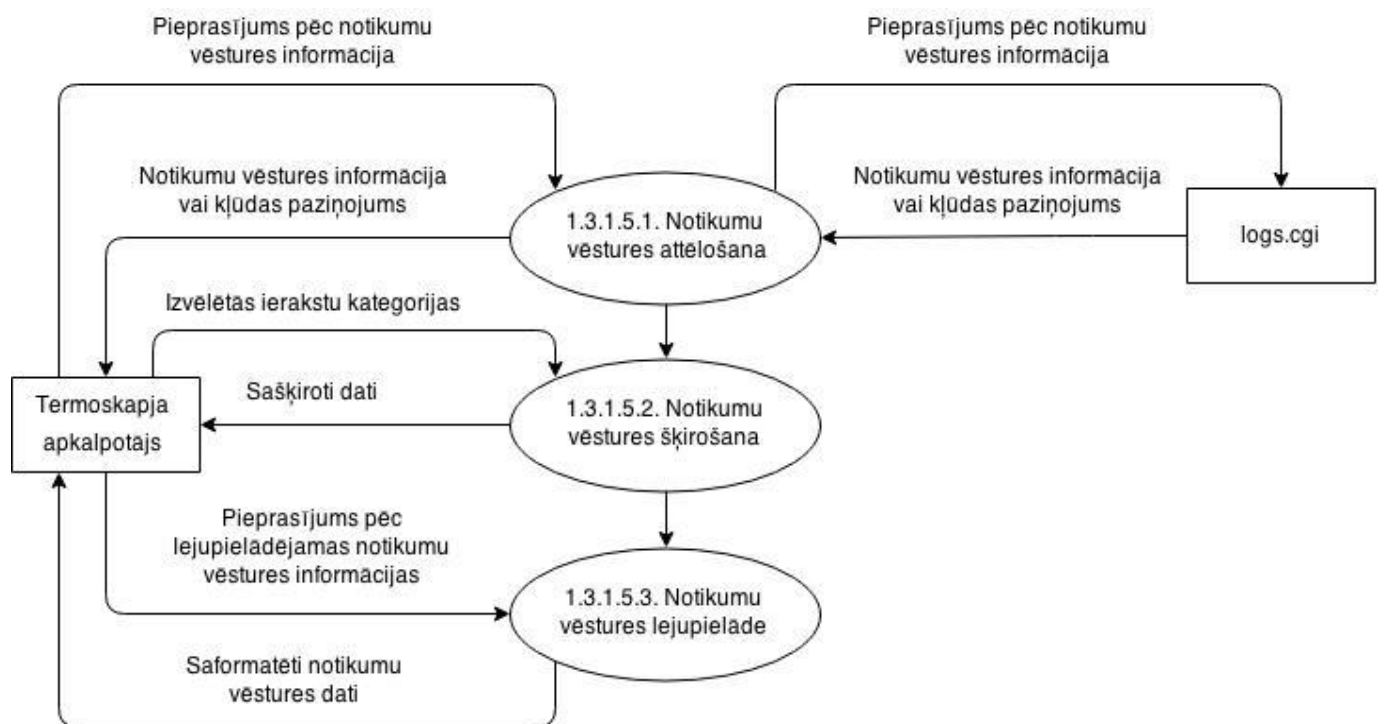


'Profiles' sadaļas 1. līmeņa DPD

2.3.1.1.5. logs.inc

Modulis ir paredzēts, lai lietotājam attēlotu notikumu vēstures informāciju pa lapaspusēm. To ir iespējams šķirot pēc notikuma ieraksta līmeņa, kā arī lejupielādēt no CSV formāta pārveidotu vieglāk lasāmā formāta.

Sīkākai informācijai skatīt nodaļas “1.3.2. Funkcionālās prasības” apakšnodaļas “1.3.2.5. Saskarnes sadaļa “History”” prasību “1.3.2.5.1. Notikumu vēstures attēlošana” un nākamās nodaļas “2.3.3. Datu dekompozīcija” sadaļu “2.3.3.5. temperature.csv”.

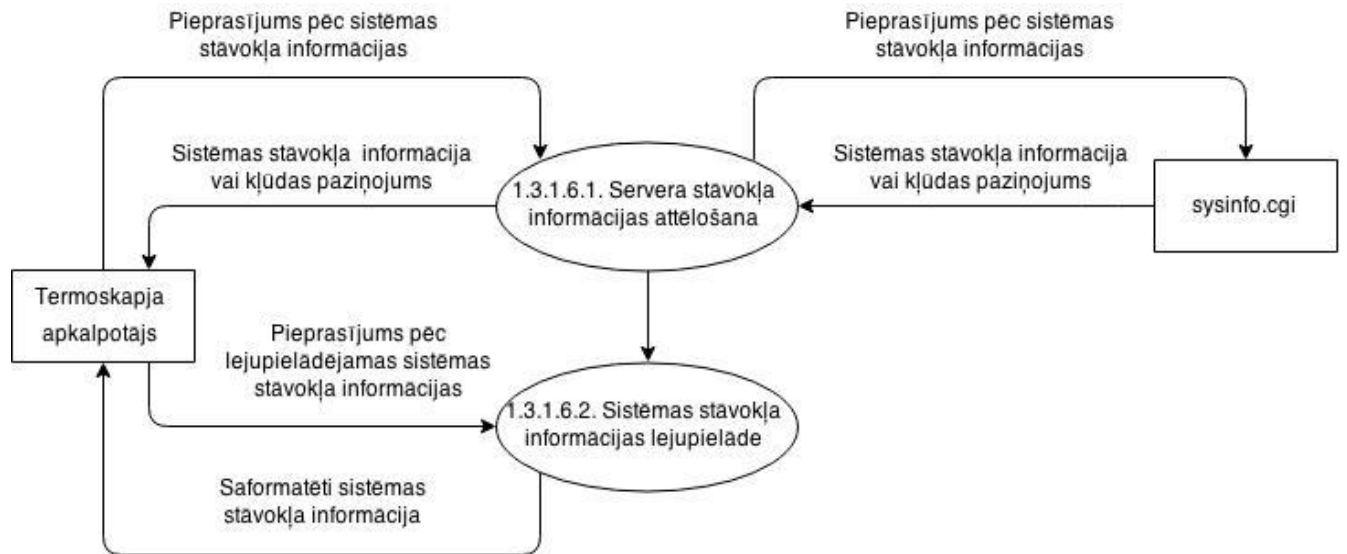


'History' sadaļa 1. līmeņa DPD

2.3.1.1.6. *sysinfo.inc*

Modulis ir paredzēts, lai lietotājam attēlotu sistēmas stāvokļa informāciju pa lapaspusēm. To ir iespējams lejupielādēt no CSV formāta pārveidotu vieglāk lasāmā formātā.

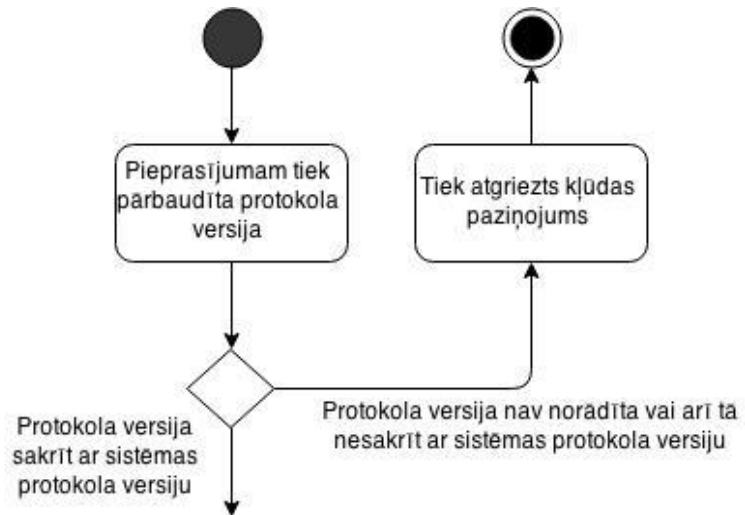
Sīkākai informācijai skatīt nodaļas “1.3.2. *Funkcionālās prasības*” apakšnodaļas “1.3.3.4. *Saskarnes sadaļa “System health”*” prasību “1.3.3.6.1. *Sistēmas stāvokļa informācijas attēlošana*” un nākamās nodaļas “2.3.3. *Datu dekompozīcija*” sadaļu “2.3.3.6. *system.csv*”.



'System health' sadaļas 1. līmeņa DPD

2.3.1.2. CGI moduļi

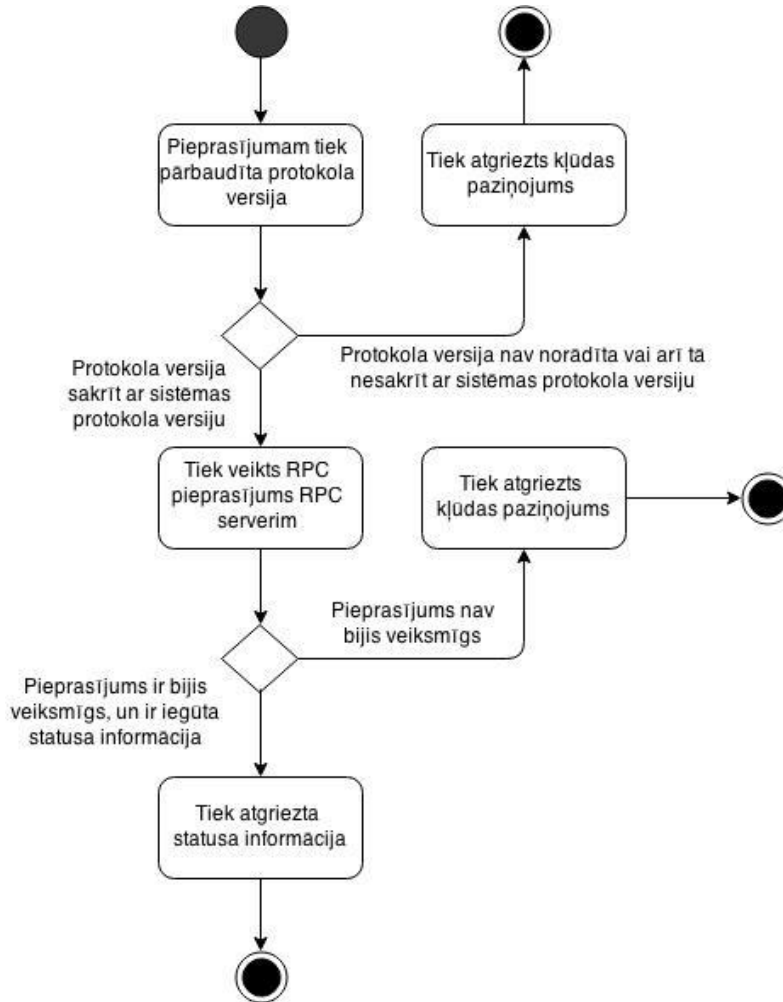
Katrs CGI modulis saņemto datu apstrādi sāk ar pieprasījuma protokola versijas pārbaudi. Lai UML aktivitāšu diagrammās izvairītos no atkārtotām, visas turpmākās UML diagrammas ir jāskata kā turpinājums UML diagrammai “*Protokola versijas pārbaude*”. Par piemēro kalpo pirmā UML aktivitāšu diagramma “*Statusa informācijas iegūšana*”.



Protokola versija pārbaude

2.3.1.2.1. *get_status.cgi*

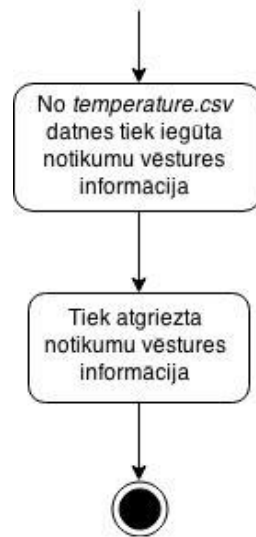
Modulis apstrādā pieprasījumus pēc sistēmas statusa informācijas – saņemot pieprasījumu no lietotāja, tiek veikts pieprasījums sistēmas RPC serverim, kas CGI modulim atgriež prasīto informāciju, ko pēc tam, sagatavojot kā atbildi, ir iespējams atgriezt lietotājam.



Statusa informācijas iegūšana

2.3.1.2.2. *get_logs.cgi*

Modulis apstrādā pieprasījumus pēc notikumu vēstures informācijas, nolasot informāciju no datnes un sagatavojot atbildi.



Notikumu vēstures informācijas iegūšana

2.3.1.2.3. *get_sysinfo.cgi*

Modulis apstrādā pieprasījumus pēc sistēmas stāvokļa informācijas, nolasot informāciju no datnes un sagatavojot atbildi.



Sistēmas stāvokļa informācijas iegūšana

2.3.1.2.4. *conf.cgi*

Modulis ir paredzēts, lai apstrādātu pieprasījumus pēc sistēmas iestatījumu informācijas iegūšanas, kā arī lai apstrādātu pieprasījumus pēc sistēmas iestatījumu maiņas, veicot ievadīto datu validāciju un nepieciešamības gadījumā sagatavojot kļūdas paziņojumus. Sistēmas iestatījumu informācijas iegūšana un rediģēšana notiek caur RPC izsaukumiem.

2.3.1.2.5. *modes.cgi*

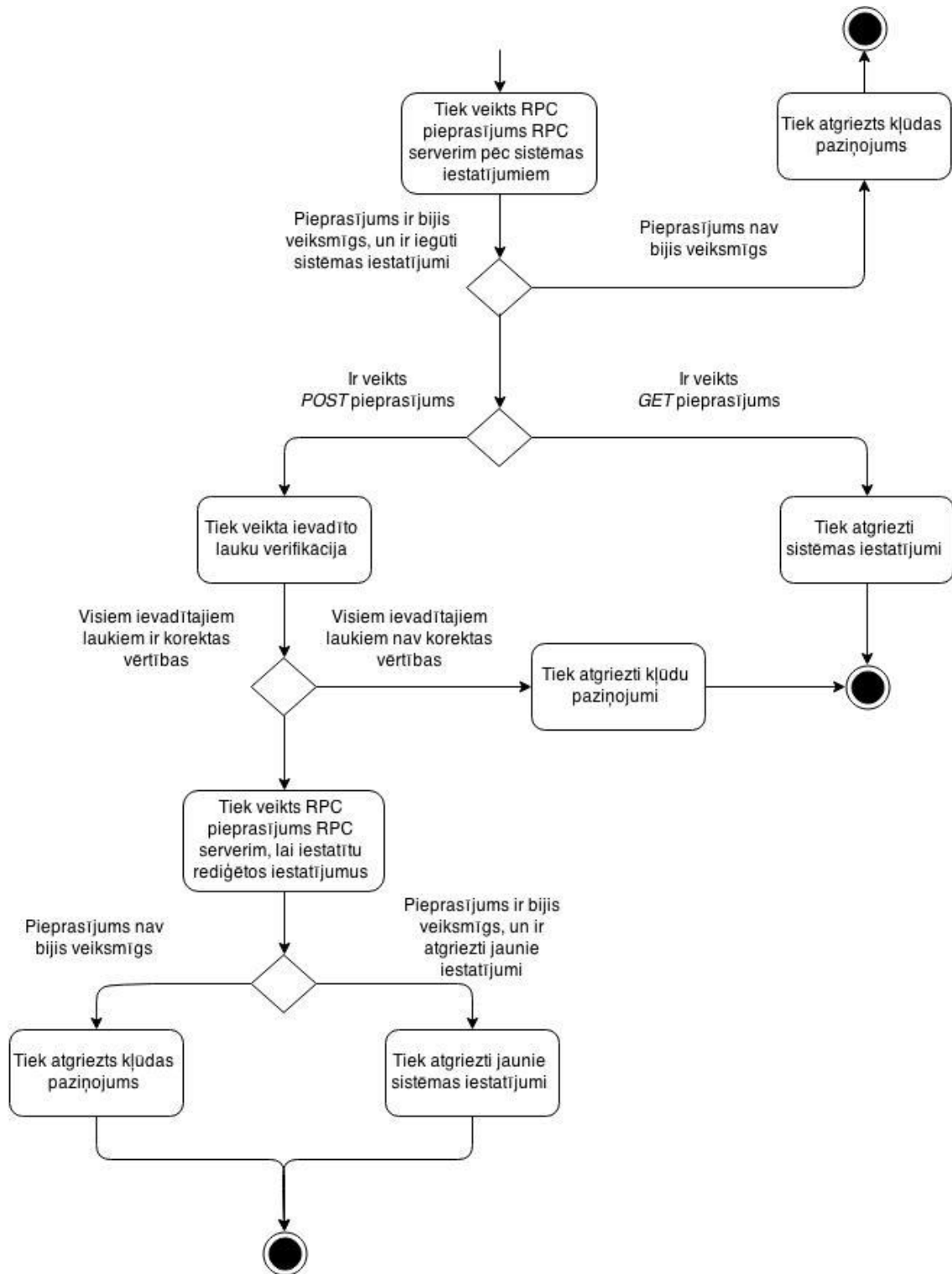
Modulis ir paredzēts, lai apstrādātu visus pieprasījumus saistībā ar termoskapja režīma maiņu. Tiek veikta ievadīto datu validācija, tai skaitā atkārtoti pārbaudot lietotāja izveidoto temperatūras profilu pirms tā nodošanas apstrādei, un nepieciešamības gadījumā sagatavoti kļūdas paziņojumi. Režīmu maiņa, stabilizācijas temperatūras un temperatūras profila nodošana apstrādei notiek caur RPC izsaukumiem.

2.3.1.2.6. *profiles.cgi*

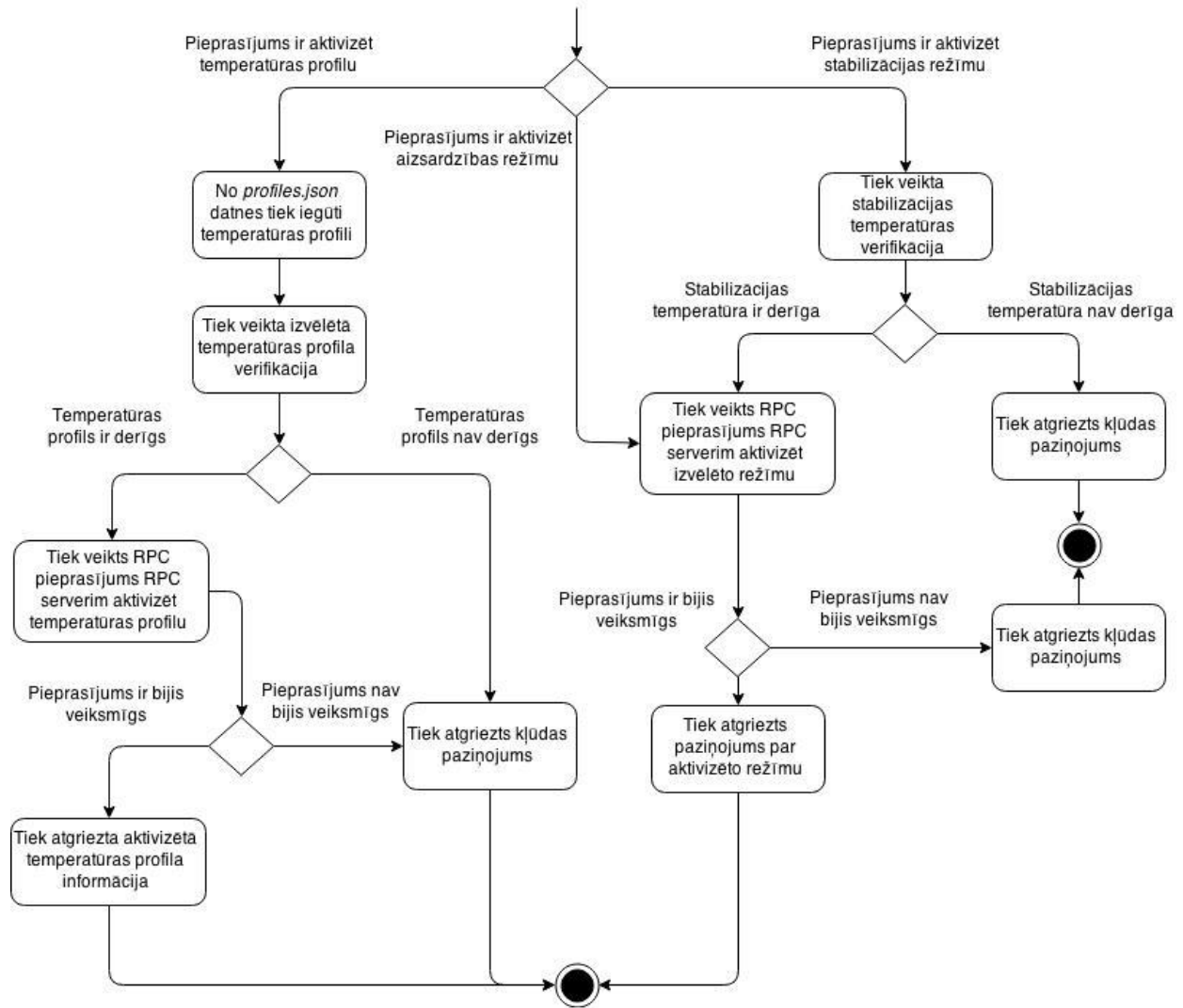
Modulis ir paredzēts, lai apstrādātu visus pieprasījumus saistībā ar termoskapja temperatūras profiliem – pieejamo temperatūras profilu iegūšanu, rediģēšanu, dzēšanu un jaunu izveidošanu. Tiek veikta ievadīto datu validācija un nepieciešamības gadījumā sagatavoti kļūdas paziņojumi. Veiksmīga pieprasījuma gadījumā tiek veikts RPC izsaukums RPC serverim, lai par temperatūras profilu izmaiņām tiktu izveidots notikumu ieraksts.

2.3.1.2.7. *interfaces.cgi*

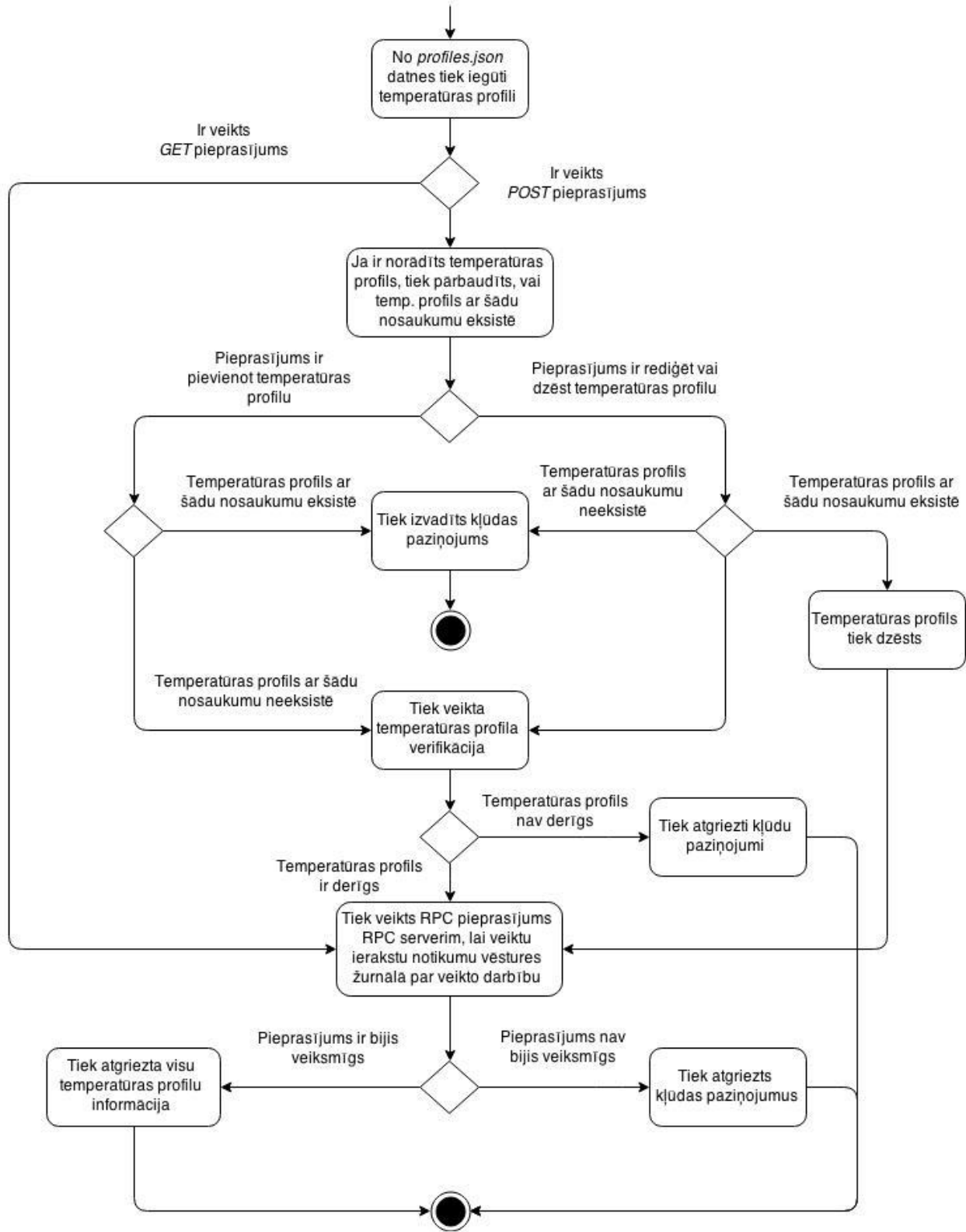
Modulis ir paredzēts, lai apstrādātu pieprasījumus pēc sistēmas interneta savienojuma iestatījumu iegūšanas un rediģēšanas, veicot dažādus komandas izsaukumus komandrindā atkarībā no pieprasījuma un esošā interneta savienojuma iestatījumiem. Tiek veikta ievadīto datu validācija un nepieciešamības gadījumā sagatavoti kļūdas paziņojumi.



Modulis 'conf.cgi'



Modulis 'modes.cgi'



Modulis 'profiles.cgi'

2.3.2. Vienlaicīgo procesu dekompozīcija

2.3.2.1. *main*

Process ir atbildīgs par sistēmas sagatavošanu darbam, ielasot un iestatot iestatījumus, statusa informāciju un citus mainīgos, par savienojuma izveidošanu ar sensoriem, kā arī par nākamajā sadaļā aprakstītā “*update_and_log*” pavediena sākšanu. Vienlaicīgi darbojas 2 vai 3 procesi, un šis process darbojas līdz sistēma tiek izslēgta.

Pēc sistēmas nepieciešamo mainīgo un datņu inicializēšanas, savienojuma izveidošanas ar sensoru, šis pavediens tiek pārvērst par RPC serveri, kas apstrādā no nodaļas “1.3.1. *Moduļu dekompozīcija*” sadaļas “1.3.1.2. *CGI moduļi*” moduļiem saņemtos RPC pieprasījumus, tos apstrādājot un sniedzot atbildi, ko pēc tam CGI modulis atgriež lietotājam.

Skatīt datu plūsmas diagrammu “Sistēmas 1. līmeņa DPD”.

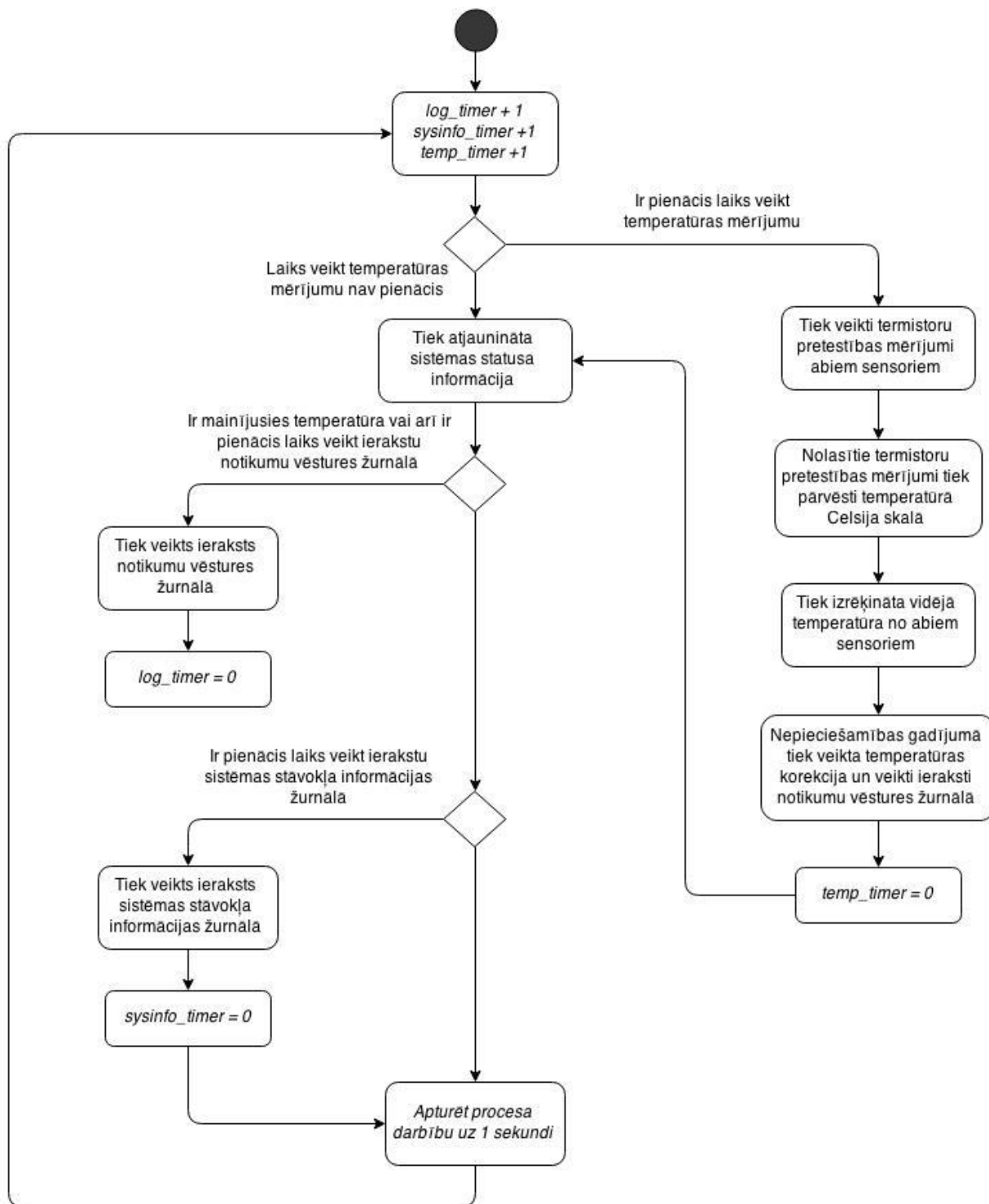
2.3.2.2. *update_and_log*

update_and_log pavediens tiek izsaukts uzreiz pēc sistēmas darbībai nepieciešamo mainīgo iestatīšanas. Tas ir atbildīgs par mērījumu un temperatūras kontroles funkciju izsaukšanu, kā arī par notikumu vēstures un sistēmas stāvokļa informācijas uzturēšanu. Tāpat kā *main* process, *update_and_log* process darbojas līdz tiek izslēgta sistēma.

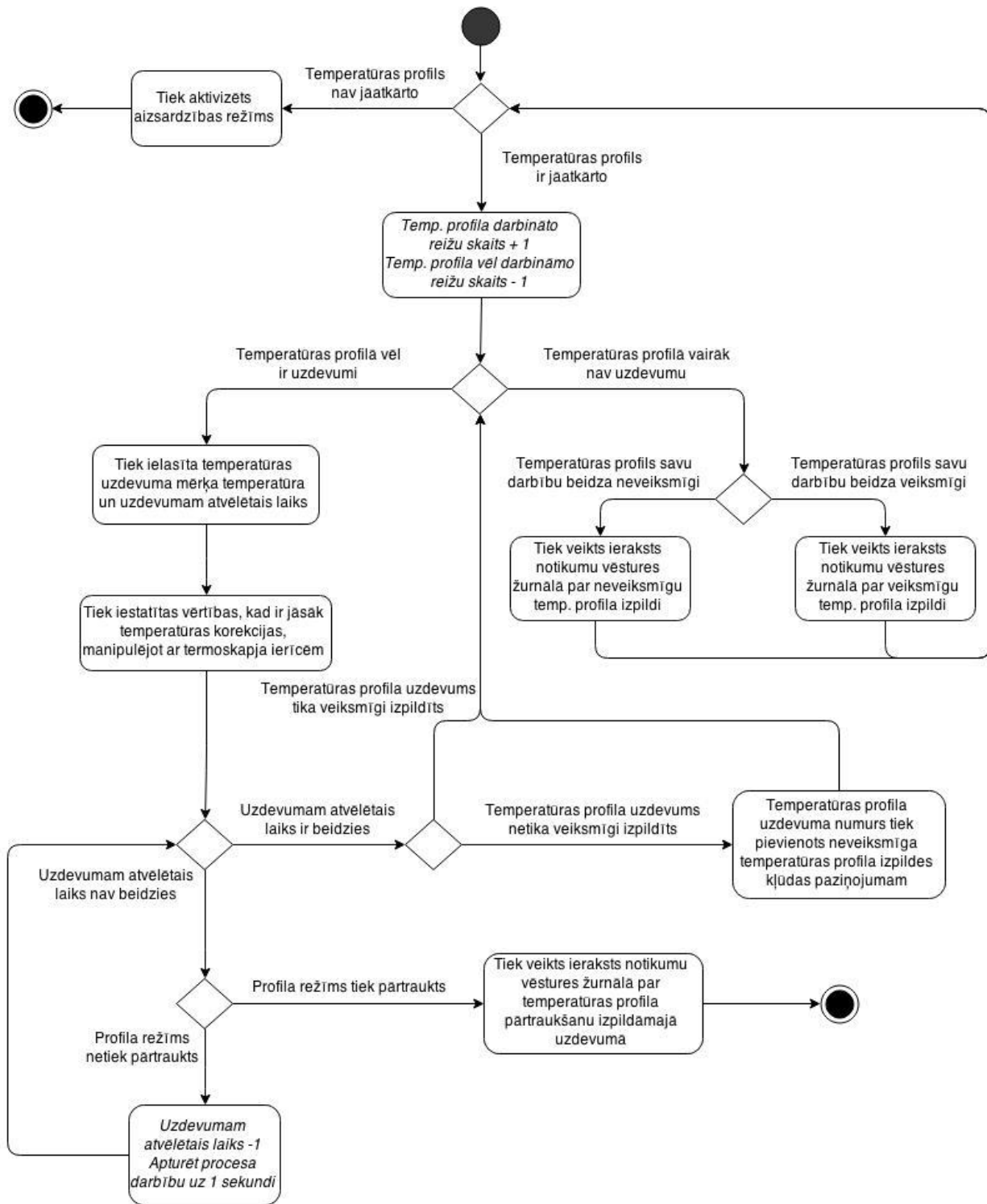
2.3.2.3. *process_profile*

process_profile pavediens tiek izsaukts pēc RPC pieprasījuma saņemšanas un apstrādes *main* pavedienā. Tas ir atbildīgs par temperatūras profila uzdevumu nolasīšanu un izpildi, kā arī par notikumu vēstures informācijas papildināšanu ar uzdevuma izpildes statusu – vai tas ir bijis veiksmīgs, vai arī ne. *process_profile* pavediens pārtrauc darbību tad, kad tiek nomainīts termoskapja režīms, vai arī temperatūras profilā ir izpildīti visi uzdevumi, kurā brīdī tiek aktivizēts aizsardzības režīms.

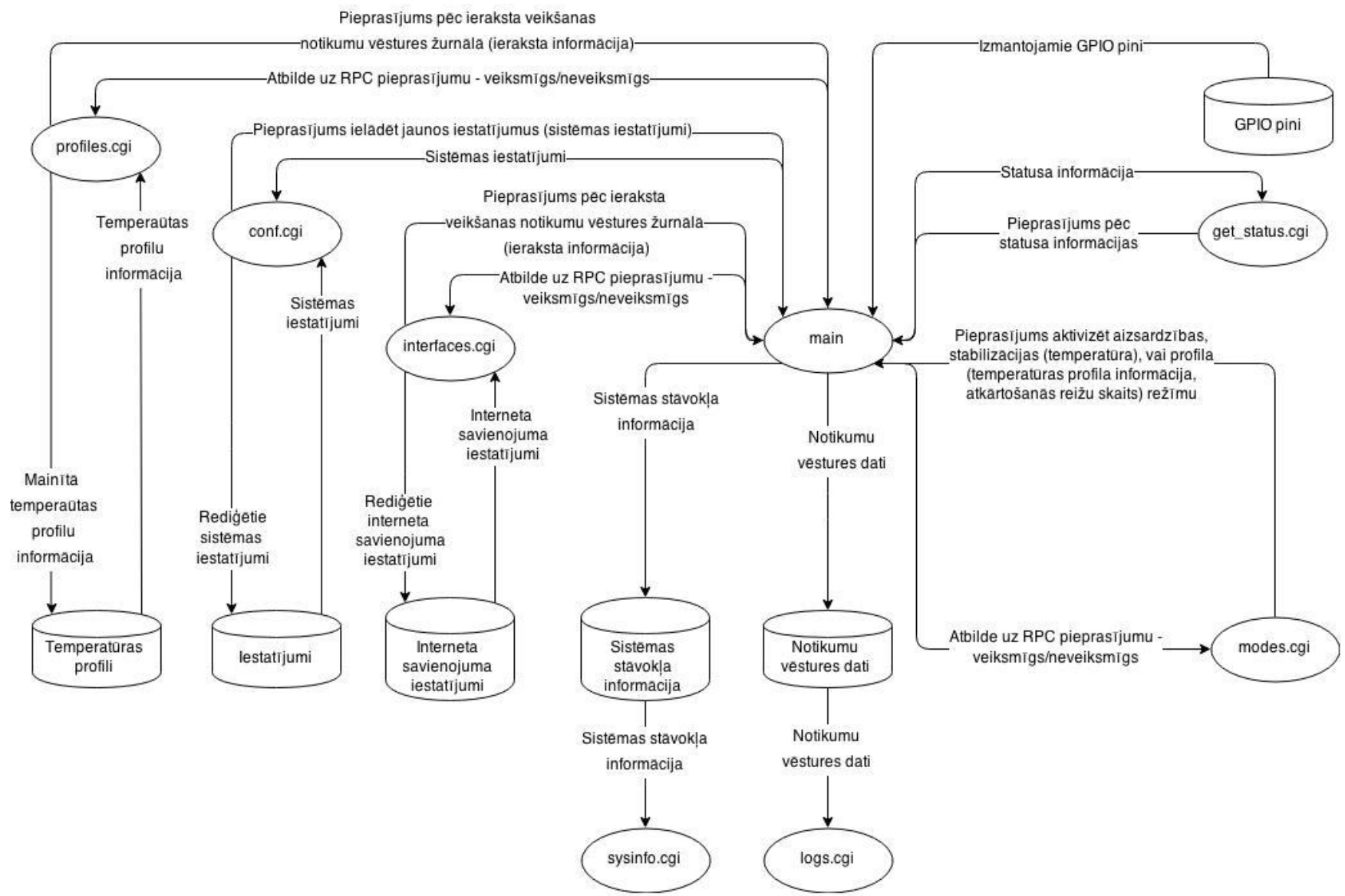
log_timer - skaitītājs, kad ir jāveic ieraksts notikumu vēstures žurnālā
sysinfo_timer - skaitītājs, kad ir jāveic ieraksts sistēmas stavokļa informācijas žurnālā
temp_timer - skaitītājs, kad ir jāveic temperatūras mērījums termoskapī



Process 'update_and_log'



Process 'process_profile'



Sistēmas 1. līmeņa DPD

2.3.3. Datu dekompozīcija

Informācija par datu laukiem – tulkojums, katra lauka nozīme un tā iespējamais vērtību diapazons – ir jau aprakstīta programmatūras prasību specifikācijas nodaļā “1.3.2. Funkcionālās prasības”. Šajā sadaļā pie konkrētajiem datiem tiks pievienota atsauce uz minēto informāciju, norādot sadaļu, demonstrēti piemēri ar pārsūtamo datu JSON struktūrām, kā arī aprakstīts notikumu vēstures un sistēmas stāvokļa informācijas moduļos izmantotā datu strukturēšana notikumu žurnālos.

2.3.3.1. *Statusa informācija*

Veiksmīga pieprasījuma atbilde no *get_status.cgi* moduļa. Informācija par laukiem un to nozīmi ir atrodama programmatūras prasību specifikācijas nodaļas “1.3.1. Funkcionālās prasības” sadaļā “1.3.1.1.1. Statusa informācijas attēlošana”.

```
{
  "protocol_ver": "0.1",
  "device_name": "TECA 2",
  "device_sn": "1234567",
  "data":
  {
    "sensors":
    [
      "-46.9",
      "-46.9"
    ],
    "uptime": "189368",
    "temperature": "-46.9",
    "target_temperature": "35.0",
    "profile": "Āfrika",
    "repeat": "1",
    "cycle": "1",
    "mode": "Profile",
    "mode_changed_by": "192.168.1.107",
    "heater": "1",
    "cooler": "0",
    "fan": "0",
    "system_time": "1434025616.707778"
  }
}
```

2.3.3.2. *Iestatījumu informācija*

Veiksmīga pieprasījuma atbilde no *conf.cgi* moduļa. Informācija par laukiem un to nozīmi ir atrodama programmatūras prasību specifikācijas nodaļas “1.3.1. Funkcionālās prasības” sadaļā “1.3.1.2.1. *Iestatījumu informācijas attēlošana attēlošana*”.

```
{
  "protocol_ver": "0.123",
  "device_name": "TECA 777",
  "device_sn": "1234567",
  "data":
  {
    "delta_on": "5",
    "delta_off": "5",
    "temperature_retrieve_period": "5",
    "status_retrieve_period": "5",
    "temperature_unit": "C",
  }
}
```

2.3.3.3. *Interneta savienojuma iestatījumu informācija*

Veiksmīga pieprasījuma atbilde no *interface.cgi* moduļa. Informācija par laukiem un to nozīmi ir atrodama programmatūras prasību specifikācijas nodaļas “1.3.1. Funkcionālās prasības” sadaļā “1.3.1.2.6. *Interneta savienojuma iestatījumu attēlošana*”.

```
{
  "address": "192.168.1.106",
  "netmask": "255.255.255.0",
  "gateway": "192.168.1.1",
  "mode": "dhcp"
}
```

2.3.3.4. *profiles.json*

Datne, kurā tiek glabāti lietotāja izveidotie temperatūras profili.

Informācija par laukiem un to nozīmi ir atrodama programmatūras prasību specifikācijas nodaļas “1.3.1. Funkcionālās prasības” sadaļā “1.3.1.3.7. Jaunu temperatūras profilu izveide”.

Temperatūras profilu darbība ir aprakstīta programmatūras prasību specifikācijas nodaļas “1.2. Vispārējs apraksts” sadaļā “1.2.1. Produkta funkcijas” un programmatūras projektējuma apraksta nodaļas “1.3.2. Vienlaicīgo procesu dekompozīcija” sadaļā “1.3.2.3. process_profile”.

```
{
  "Āfrika":
  {
    "title":"Āfrika",
    "description":"Augsta temperatūra",
    "tasks":
    {
      "task_1":
      {
        "time":"120",
        "temperature":"35"
      },
      "task_2":
      {
        "time":"360",
        "temperature":"35"
      },
      "task_3":
      {
        "time":"30",
        "temperature":"40"
      },
      "task_4":
      {
        "time":"60",
        "temperature":"40"
      },
      "task_5":
      {
        "time":"30",
        "temperature":"45"
      },
      "task_6":
      {
        "time":"60",
        "temperature":"45"
      }
    }
  },
}
```

```

"Kanāda":
{
  "title":"Kanāda",
  "description":"Zema temperatūra",
  "tasks":
  {
    "task_1":
    {
      "time":"120",
      "temperature":"-35"
    },
    "task_2":
    {
      "time":"360",
      "temperature":"-35"
    },
    "task_3":
    {
      "time":"30",
      "temperature":"-40"
    },
    "task_4":
    {
      "time":"60",
      "temperature":"-40"
    },
    "task_5":
    {
      "time":"30",
      "temperature":"-45"
    },
    "task_6":
    {
      "time":"360",
      "temperature":"-45"
    }
  }
}
}

```

2.3.3.5. *temperature.csv*

Datne, kurā tiek glabāti notikumu vēstures dati CSV formātā, lai ietaupītu atmiņu. Tā kā katrs ieraksts ir kompakts, nesaturot neko lieku, tiek ietaupīts arī informācijas rakstīšanai atvēlētais laiks.

CSV kolonas ir sadalītas sekojoši: (sīkāka informācija aprakstīta programmatūras prasību specifikācijas nodaļas “1.3.1. Funkcionālās prasības” sadaļā “1.3.1.5.2. *Notikumu vēstures attēlošana*”):

- notikumu vēstures ieraksta numurs;
- laiks sekundēs kopš sistēmas ieslēgšanas brīža;
- servera datums un laiks UNIX sekundēs;
- notikumu vēstures ieraksta līmenis (iespējamie ierakstu līmeņi:
 - 10 – kļūda;
 - 6 – brīdinājums;
 - 5 – atjauninājums (iestatījumu vai temperatūras profila);
 - 3 – informācija (ieslēgts režīms, pabeigts temperatūras profila uzdevums);
 - 0 – statusa informācija.);
- ieraksta vai darbības veicējs;
- statusa informācija (vērtību nozīme:
 - 1. vērtība – “1”, ja ir ieslēgts sildītājs, “0”, ja tas ir izslēgts;
 - 2. vērtība – “1”, ja ir ieslēgts dzesētājs, “0”, ja tas ir izslēgts;
 - 3. vērtība – “1”, ja ir ieslēgts ventilators, “0”, ja tas ir izslēgts;
 - 4. vērtība – no sensoru rādītājiem izrēķinātā vidējā temperatūra termoskapī Celsija skalā;
 - 5. vērtība – mērķa temperatūra ieraksta brīdī Celsija skalā.);
- paziņojums.

Piemērs no notikumu vēstures datiem:

```
4547,3041,1426066457,5,192.168.1.81,0/0/0/-5.5/-35.0,Changed configuration.;
```

2.3.3.6. *system.csv*

Tāpat kā notikumu vēstures informācija, arī sistēmas stāvokļa informācija tiek glabāta datnē CSV formātā.

CSV kolonas ir sadalītas sekojoši (sīkāka informācija aprakstīta programmatūras prasību specifikācijas nodaļas “1.3.1. Funkcionālās prasības” sadaļā “1.3.1.6.2. *Notikumu vēstures attēlošana*”):

- sistēmas stāvokļa ieraksta numurs;
- laiks sekundēs kopš sistēmas ieslēgšanas brīža;
- servera datums un laiks UNIX sekundēs;
- sistēmas patērētais operatīvās atmiņas daudzums kilobaitos;
- sistēmai pieejamā operatīvā atmiņa kilobaitos;
- sistēmai patērētie procesora resursi no kopējiem resursiem (procentuāli);
- kopējā servera procesora noslodze;
- serverī notiekošo procesu skaits.

2.4. Trasējamības tabula

2.4.1. Saskaņes sadaļa “*Status*”

Funkcionālā prasība	Projektētā metode
1.3.1.1.1. Statusa informācijas attēlošana	displayStatus
1.3.1.1.2. Termoskapja temperatūras vēstures attēlošana grafikā	displayGraph
1.3.1.1.3. Tekošās termoskapja temperatūras attēlošana grafikā	liveUpdate
1.3.1.1.4. Informācijas attēlošana par temperatūras punktu grafikā	showPointInformation

2.4.2. Saskaņes sadaļa “*Configuration*”

Funkcionālā prasība	Projektētā metode
1.3.1.2.1. Iestatījumu informācijas attēlošana	displayConf
1.3.1.2.2. Iestatījumu rediģēšana	set_conf
1.3.1.2.3. Sistēmas laika lauka aizpilde ar lietotāja datora laiku	setMyTime
1.3.1.2.4. Interneta savienojuma iestatījumu attēlošana	displayNetwork
1.3.1.2.5. Interneta savienojuma iestatījumu rediģēšana	set_interface

2.4.3. Sāskarnes sadaļa “Modes”

Funkcionālā prasība	Projektētā metode
1.3.1.3.1. Iespējamo režīmu un temperatūras profilu attēlošana	createStabilizationOptions, displayProfiles
1.3.1.3.2. Paziņojumu attēlošana izvēloties aizsardzības režīmu	createAppropriateView, alertUser
1.3.1.3.3. Aizsardzības režīma aktivizēšana	startMode
1.3.1.3.4. Paziņojumu attēlošana izvēloties stabilizācijas režīmu	createAppropriateView, alertUser
1.3.1.3.5. Stabilizācijas režīma aktivizēšana	startMode
1.3.1.3.6. Paziņojumu attēlošana izvēloties temperatūras profilu	createAppropriateView, alertUser
1.3.1.3.7. Temperatūras profila aktivizēšana	startMode

2.4.4. Sāskarnes sadaļa “Profiles”

Funkcionālā prasība	Projektētā metode
1.3.1.4.1. Temperatūras profilu attēlošana	displayProfiles
1.3.1.4.2. Temperatūras profila uzdevumu attēlošana	showSelectedProfileInfo
1.3.1.4.3. Temperatūras profila grafika attēlošana	drawProfile
1.3.1.4.4. Informācijas attēlošana par temperatūras punktu grafikā	showPointInformation
1.3.1.4.5. Jauna temperatūras profila izveide	add_profile
1.3.1.4.6. Jauna temperatūras profila izveide, par pamatu izvēloties esošu profilu	useAsTemplate
1.3.1.4.7. Temperatūra profila rediģēšana	edit_profile
1.3.1.4.8. Temperatūras profila dzēšana	remove_profile

2.4.5. Sāskarnes sadaļa “History”

Funkcionālā prasība	Projektētā metode
1.3.1.5.1. Notikumu vēstures attēlošana	displayLogs
1.3.1.5.2. Notikumu vēstures šķirošana	showOrHideErrorEntries, showOrHideWarningEntries, showOrHideUpdateEntries, showOrHideInfoEntries, showOrHideDebugEntries
1.3.1.5.3. Notikumu vēstures datu lejupielāde	downloadLogs

2.4.6. Sāskarnes sadaļa “System health”

Funkcionālā prasība	Projektētā metode
1.3.1.6.1. Sistēmas stāvokļa informācijas attēlošana	displayLogs
1.3.1.6.2. Sistēmas stāvokļa informācijas lejupielāde	downloadLogs

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

Lai atvieglotu, paātrinātu un sistematizētu termoskapju vadības sistēmas saskarnes vienībtestēšanu, tika izstrādāti automatizētie vienībtesti JavaScript programmēšanas valodā, izmantojot Selenium programmatūras testēšanas ietvaru.

Testēšanas rezultāti kā citās dokumentācijas sadaļās ir apkopoti pa saskarnes sadaļām.

Testēšanas dokumentācijā ir aprakstītas veiktās pārbaudes, kuras ir apkopotas no katras konkrēto prasību saskarnes sadaļas nodaļas programmatūras prasību specifikācijā rindas kārtībā, un to rezultāti. Mērķis bija pārliecināties, ka testējamo vienumu darbība atbilst programmatūras prasību specifikācijā izvirzītajām prasībām.

Testēšanas rezultātu tabulās ir arī vienībtesti, kas ir veikti manuāli, kuru neizpilde kādā brīdī nākotnē ir apšaubāma un kuru izstrādei patērētā laika daudzums pret ieguvumu būtu pārāk mazs. Pie šiem vienībtestiem iekavās ir pievienots burts “M”, apzīmējot vārdu *manual*.

Saskarnes sadaļas “*Status*” un “*Modes*” ir apskatītas kopā, jo abas testēšanas nolūkos ir savstarpēji saistītas.

Testēšana tika veikta, izmantojot pārlūkprogrammu Mozilla FireFox 37.0.

3.2. Testēšanas rezultāti

3.2.1. “*Status*” un “*Modes*” sadaļas

Pārbaude	Izpildās
Lauka “ <i>Uptime</i> ” vērtība tiek attēlota formātā <i>A weeks B days HH:MM:SS</i> (kur A ir nedēļas, B dienas). (M)	Jā
Nospiežot pogu “ <i>Live update is ON</i> ”, tā pārvēršas par pogu “ <i>Live update is OFF</i> ”. (M)	Jā

3.2.1.1. Tests “Tiek attēloti korekti lauki, korektas lauku vērtības, paziņojumi un pogas aizsardzības režīmā.”

“Status” sadaļa

Pārbaude	Izpildās
Lietotājam tiek attēloti šādi lauki: <ul style="list-style-type: none"> • <i>Device name</i> • <i>Uptime</i> • <i>Sensors</i> • <i>Temperature</i> • <i>Mode</i> • <i>Mode changed by</i> • <i>Heater</i> • <i>Cooler</i> • <i>Fan</i> 	Jā
Lietotājam netiek attēloti šādi lauki: <ul style="list-style-type: none"> • <i>Target temperature</i> • <i>Profile</i> 	Jā
Lietotājam tiek attēlotas šādas pogas: <ul style="list-style-type: none"> • “<i>Live update is ON</i>” 	Jā
Lietotājam netiek attēlotas šādas pogas: <ul style="list-style-type: none"> • “<i>Protect</i>” 	Jā

“Modes” sadaļa

Pārbaude	Izpildās
Ja aizsardzības režīms jau ir aktīvs, lietotājam tiek attēlots paziņojums “Protection mode already active.” un poga “Proceed”, lai aktivizētu izvēlēto režīmu ir izslēgta.	Jā
Ja ir aktīvs stabilizācijas režīms, lietotājam tiek attēlots paziņojums: “About to activate protection mode. Stabilization at A B degrees will be stopped.” (kur A ir izvēlēta stabilizācijas temperatūra un B lietotāja iestatītā temperatūras skala).	Jā
Ja ir aktīvs temperatūras profils, lietotājam tiek attēlots paziņojums: “About to activate protection mode. Profile “A” on cycle B from C will be stopped.” (kur A ir temperatūras profila nosaukums, B – cik reizes temperatūras profils ir atkārots, C –	Jā

cik reizes temperatūras profils ir jāatkārto kopā.)	
Veiksmīga pieprasījuma gadījumā lietotājam tiek attēlots paziņojums: <i>“Protection mode has been activated.”</i>	Jā

3.2.1.2. Tests “Tiek attēloti korekti lauki, korektas lauku vērtības, paziņojumi un pogas stabilizācijas režīmā.”

“Status” sadaļa

Pārbaude	Izpildās
Lietotājam tiek attēloti šādi lauki: <ul style="list-style-type: none"> • <i>Device name</i> • <i>Uptime</i> • <i>Sensors</i> • <i>Temperature</i> • <i>Target temperature</i> • <i>Mode</i> • <i>Mode changed by</i> • <i>Heater</i> • <i>Cooler</i> • <i>Fan</i> 	Jā
Lietotājam netiek attēloti šādi lauki: <ul style="list-style-type: none"> • <i>Profile</i> 	Jā
Lietotājam tiek attēlotas šādas pogas: <ul style="list-style-type: none"> • <i>“Live update is ON”</i> • <i>“Protect”</i> 	Jā

“Modes” sadaļa

Pārbaude	Izpildās
Izvēloties stabilizācijas režīmu, lietotājam tiek attēlots paziņojums: <i>“Either select or enter the stabilization temperature.”</i> un poga <i>“Proceed”</i> nav iespējams nospiegt, kamēr nav izvēlēta stabilizācijas temperatūra.	Jā
Izvēloties stabilizācijas režīmu no izvēles iespējām vai arī ievadot temperatūru ievadlaukā, lietotājam tiek attēlots paziņojums: <i>“About to activate stabilization mode at A B degrees.”</i> (kur A ir	Jā

stabilizācijas temperatūra, un B lietotāja izvēlētā temperatūras skala).	
Ja stabilizācijas režīms jau ir aktīvs, lietotājam papildus vienam no augstākminētajiem paziņojumiem tiek attēlots paziņojums “ <i>Stabilization at A B degrees will be stopped.</i> ” (kur A ir stabilizācijas temperatūra, un B lietotāja izvēlētā temperatūras skala).	Jā
Ja ir aktīvs temperatūras profils, lietotājam tiek attēlots paziņojums: “ <i>Profile "A" on cycle B from C will be stopped.</i> ” (kur A ir temperatūras profila nosaukums, B – cik reizes temperatūras profils ir atkārots, C – cik reizes temperatūras profils ir jāatkārto kopā.).	Jā
Nekorektu ievaddatu gadījumā lietotājam tiek attēloti kļūdu paziņojumi, norādot uz konkrētiem laukiem, kas nav bijuši korekti aizpildīti un to iespējamajām vērtībām.	Jā
Veiksmīga pieprasījuma gadījumā lietotājam tiek attēlots paziņojums: “ <i>Stabilization at A B degrees has been activated.</i> ” (Kur A ir stabilizācijas temperatūra B temperatūras skala.)	Jā

3.2.1.1. Tests “Tiek attēloti korekti lauki, korektas lauku vērtības, paziņojumi un pogas temperatūras profila režīmā.”

“Status” sadaļa

Pārbaude	Izpildās
<p>Lietotājam tiek attēloti šādi lauki:</p> <ul style="list-style-type: none"> • <i>Device name</i> • <i>Uptime</i> • <i>Sensors</i> • <i>Temperature</i> • <i>Target temperature</i> • <i>Mode</i> • <i>Profile</i> • <i>Mode changed by</i> • <i>Heater</i> • <i>Cooler</i> • <i>Fan</i> 	Jā
<p>Lietotājam tiek attēlotas šādas pogas:</p> <ul style="list-style-type: none"> • “<i>Live update is ON</i>” • “<i>Protect</i>” 	Jā

“Modes” sadaļa

Pārbaude	Izpildās
<p>Izvēloties profilu režīmu, lietotājam tiek attēlots paziņojums “<i>Select the profile to activate.</i>” un pogu “<i>Proceed</i>” nav iespējams nospiegt, kamēr nav izvēlēts temperatūras profils un temperatūras profila atkārtotās reizes skaits.</p>	Jā

<p>Izvēloties temperatūras profilu no izvēlnes parādās ievadlauks, kur ievadīt profila atkārtošanās reižu skaitu (0, lai profilu atkārtotu līdz režīma maiņai) un lietotājam tiek attēlots vai nu paziņojums</p> <ul style="list-style-type: none"> • <i>“About to activate profile “A” to repeat B times.”</i> (kur A ir temperatūras profila nosaukums, B – cik reizes temperatūras profilu atkārtos), vai nu paziņojums • <i>“About to activate profile “A” to repeat till stopped.”</i> (kur A ir temperatūras profila nosaukums), ja temperatūras profila atkārtošanās reižu skaita ievadlaukā ir ievadīta nulle. 	<p>Jā</p>
<p>Ja jau ir aktīvs temperatūras profils vai arī stabilizācijas režīmam, lietotājam papildus vienam no augstākminētajiem paziņojumiem tiek attēlots vai nu paziņojums</p> <ul style="list-style-type: none"> • <i>“Profile “A” on cycle B from C will be stopped.”</i> (kur A ir temperatūras profila nosaukums, B – cik reizes temperatūras profils ir atkārots, C – cik reizes temperatūras profils ir jāatkārto kopā.), vai nu paziņojums • <i>“Profile “A” will be stopped.”</i> (kur A ir temperatūras profils), ja temperatūras profils darbojas līdz režīma maiņai, vai nu paziņojums, vai nu paziņojums • <i>“Stabilization at A B degrees will be stopped.”</i> (kur A ir stabilizācijas temperatūra, un B lietotāja izvēlēta temperatūras skala). 	<p>Jā</p>
<p>Nekorektu ievaddatu gadījumā lietotājam tiek attēloti kļūdu paziņojumi, norādot uz konkrētiem laukiem, kas nav bijuši korekti aizpildīti un to iespējamajām vērtībām.</p>	<p>Jā</p>

3.2.2. “Configuration” sadaļa

Pārbaude	Izpildās
Lietotājam tiek attēloti šādi iestatījumu lauki: <ul style="list-style-type: none"> • <i>Protocol version</i> • <i>Device name</i> • <i>Device serial number</i> • <i>Delta on</i> • <i>Delta off</i> • <i>Temperature unit</i> • <i>Temperature retrieve period</i> • <i>Status retrieve period</i> • <i>System time</i> 	Jā
Lauka “ <i>System time</i> ” vērtība tiek formatēts formātā YYYY-MM-DD HH-MM-SS. (M)	Jā
Lietotājam ir iespējams rediģēt šādus sistēmas iestatījumu laukus: <ul style="list-style-type: none"> • <i>Device name</i> • <i>Device serial number</i> • <i>Delta on</i> • <i>Delta off</i> • <i>Temperature unit</i> • <i>Temperature retrieve period</i> • <i>Status retrieve period</i> • <i>System time</i> 	Jā
Lietotājam nav iespēja rediģēt lauku “ <i>Protocol version</i> ”.	Jā
Ir iespējams iestatīt trīs temperatūras skalas: Celsija, Fārenheita un Kelvina.	Jā
Nospiežot pogu “ <i>Set my time</i> ”, “ <i>System time</i> ” lauka vērtība tiek aizpildīta ar lietotāja datora laiku YYYY-MM-DD HH-MM-SS formātā.	Jā
Lauku “ <i>Delta on</i> ” un “ <i>Delta off</i> ” vērtības pirms nosūtīšanas serverim tiek pārvērstas Celsija grādos. (M)	Jā
Ievadītais sistēmas laiks pirms nosūtīšanas serverim tiek pārvērsts par UNIX sekundēm. (M)	Jā

<p>Serveris veic datu verifikāciju:</p> <ul style="list-style-type: none"> • vai ir ievadīts termoskapja nosaukums; • vai ir ievadīta <i>delta on</i> vērtība, un vai tas ir skaitlis robežā no 1 līdz 10; • vai ir ievadīta <i>delta off</i> vērtība, un vai tas ir skaitlis robežā no 1 līdz 10; • vai ir izvēlēta viena no pieejamajām temperatūras attēlošanas skalām – C (Celsija), F (Fārenheita) vai K (Kelvina); • vai ir ievadīts temperatūras iegūšanas intervāls, un vai tas ir skaitlis robežā no 1 līdz 60, • vai ir ievadīts statusa informācijas iegūšanas intervāls, un vai tas ir skaitlis robežā no 1 līdz 300 (5 minūtes); • vai ir ievadīts sistēmas laiks. 	Jā
<p>Korektu ievaddatu gadījumā, lietotājam tiek attēlots paziņojums “<i>The new configuration has been saved.</i>”, un iestatījumu lauku vērtības tiek atjauninātas uz nomainītajām.</p>	Jā
<p>Nekorektu ievaddatu gadījumā lietotājam tiek attēloti kļūdu paziņojumi, norādot uz konkrētiem laukiem, kas nav bijuši korekti aizpildīti un to iespējamajām vērtībām.</p>	Jā
<p>Lietotājam tiek attēloti šādi interneta savienojuma iestatījumu lauki:</p> <ul style="list-style-type: none"> • <i>Mode</i> • <i>Address</i> • <i>Netmask</i> • <i>Gateway</i> 	Jā
<p>Lietotājam ir iespējams pāriet “<i>Static IP</i>” režīmā. (M)</p>	Jā

<p>Lietotājam ir iespējams rediģēt šādus sistēmas interneta savienojuma iestatījumus “<i>Static IP</i>” režīmā:</p> <ul style="list-style-type: none"> • <i>Address</i> • <i>Netmask</i> • <i>Gateway</i> 	Jā
<p>Lietotājam ir iespējams pāriet “<i>DHCP</i>” režīmā. (M)</p>	Jā
<p>Interneta savienojuma iestatījumu lauki:</p> <ul style="list-style-type: none"> • <i>Address</i> • <i>Netmask</i> • <i>Gateway</i> <p>nav labojami “<i>DHCP</i>” režīmā.</p>	Jā.
<p>Korektu izmaiņu gadījumā, lietotājam tiek attēlots paziņojums “<i>The network interface was updated successfully.</i>”, un iestatījumu lauku vērtības tiek atjauninātas uz jaunajām vērtībām.</p>	Jā
<p>Nekorektu ievaddatu gadījumā lietotājam tiek attēloti kļūdu paziņojumi, norādot uz konkrētiem laukiem, kas nav bijuši korekti aizpildīti un to iespējamajām vērtībām.</p>	Jā

3.1.3. “Profiles” sadaļa

Pārbaude	Izpildās
<i>Temperatūras profili tiek attēloti.</i>	Jā
Temperatūras profila uzdevumi tiek attēloti zem pieejamo profilu izvēlnes kā saraksts no 1 līdz N, kur N ir konkrētā temperatūras profila uzdevumu skaits. (M)	Jā
Temperatūra temperatūras profila uzdevumā tiek attēlota lietotāja iestatījumos iestatītajā temperatūrā. (M)	Jā
Temperatūra temperatūras profila uzdevumā tiek attēlota ar precizitāti līdz desmitdaļai. (M)	Jā
Temperatūras profila uzdevumi tiek apstrādāti un attēloti grafikā kā punkti līknē. (M)	Jā
Grafiks tiek atjaunots ar katru veikto izmaiņu uzdevumā vai arī ar katru pievienoto jauno uzdevuma temperatūras un uzdevumam paredzētā laika pāri. (M)	Jā
Lietotājam zem grafika tiek attēlots lauks “ <i>Temperature</i> ” ar konkrētā temperatūras profila uzdevuma temperatūru un lauks “ <i>Time</i> ” labošanas skatā ar vērtību, pēc cik minūtēm temperatūra tiktu sasniegta. Temperatūras profilu izvēles skatā lauka “ <i>Time</i> ” vietā tiek attēlots lauks “ <i>Date</i> ”, kas pretēji “ <i>Time</i> ” laukam parādītu relatīvo laiku – datumu un laiku, kad temperatūras profila uzdevuma temperatūra tiktu prognozēti sasniegta. (M)	Jā
Attēlojot informāciju par temperatūras punktu, temperatūra tiek attēlota lietotāja iestatījumos iestatītajā temperatūras skalā. (M)	Jā
Attēlojot informāciju par temperatūras punktu, temperatūras vērtībām visās temperatūras skalās tiek piemērota precizitāte ar vienu ciparu aiz komata. (M)	Jā

Attēlojot informāciju par temperatūras punktu, "Date" lauka vērtība profilu izvēles skatā attēlota YYYY-MM-DD HH-MM-SS formātā. (M)	Jā
Izveidojot temperatūras profilu, korektu ievaddatu gadījumā, lietotājam tiek attēlots paziņojums: "Profile "A" has been added." (kur A ir temperatūras profila nosaukums).	Jā
Veidojot un labojot temperatūras profilu, nekorektu uzdevuma temperatūras un laika gadījumā, lietotājam tiek attēloti paziņojumi: <ul style="list-style-type: none"> • "Invalid temperature. Possible values [-50.0;55.0] (C), [-58.0;131.0] (F), [223.1;328.1] (K)."; • "Invalid time. Possible values [1;2000].". 	Jā
Veidojot un labojot temperatūras profilu, pievienojot uzdevumu bez vērtībām, lietotāja tiek attēloti paziņojumi: <ul style="list-style-type: none"> • "Temperature value cannot be empty."; • "Time value cannot be empty.". 	Jā
Temperatūras profila uzdevumu temperatūra pirms nosūtīšanas serverim tiek pārveidota Celsija skalā. (M)	Jā
Nav iespējams izveidot temperatūras profilu bez nosaukuma. Lietotājam tiek attēlots paziņojums: "Enter a profile title or select a profile."	Jā
Iespējams izveidot jaunu temperatūras profilu, par pamatu izvēloties esošu profilu.	Jā
Labojot temperatūras profilu, korektu ievaddatu gadījumā, lietotājam tiek attēlots paziņojums: "Profile "A" has been successfully edited."	Jā
Dzēšot temperatūras profilu, veiksmīga pieprasījuma gadījumā, lietotājam tiek attēlots paziņojums: "Profile "A" has been removed." (kur A ir profila nosaukums, kas tika izdzēsts).	Jā

3.1.4. “History” sadaļa

Pārbaude	Izpildās
Notikumu vēstures informācija korekti tiek attēlota sekojošās kolonās (M): <ul style="list-style-type: none"> • <i>Nr</i> • <i>Uptime</i> • <i>Local time</i> • <i>Logging level</i> • <i>Source</i> • <i>Message</i> 	Jā
Kolonas “ <i>Uptime</i> ” vērtība nepieciešamības gadījumā tiek pārveidota līdz nedēļām. (M)	Jā
Kolonas “ <i>Local time</i> ” vērtība tiek attēlota YYYY-MM-DD HH-MM-SS formātā. (M)	Jā

3.1.5. “System health” sadaļa

Pārbaude	Izpildās
Sistēmas stāvokļa informācija korekti tiek attēlota sekojošās kolonās (M): <ul style="list-style-type: none"> • <i>Nr</i> • <i>Uptime</i> • <i>Local time</i> • <i>Daemon RAM (KB)</i> • <i>Free RAM (KB)</i> • <i>Daemon CPU (%)</i> • <i>Total CPU (%)</i> • <i>Processes</i> 	Jā
Kolonas “ <i>Uptime</i> ” vērtība nepieciešamības gadījumā tiek pārveidota līdz nedēļām. (M)	Jā
Kolonas “ <i>Local time</i> ” vērtība tiek attēlota YYYY-MM-DD HH-MM-SS formātā. (M)	Jā

3.2. Secinājumi

Izstrādātās sistēmas testešana programmatūras prasību specifikācijā izvirzītajām prasībām noritēja veiksmīgi.

Sistēmas izstrādātājs pārliecinājās par automatizēto vienībtestu pienesumu projektu izstrādei. Ja automatizētie vienībtesti būtu izstrādāti ātrāk, piemēram, vienlaikus ar funkcionalitātes izstrādi, un tiktu izmantoti kā regresa testi turpmākai izstrādei, tad šāda pieeja ievērojami paaugstinātu efektivitāti un samazinātu nepilnību vai kļūdu labošanai patērēto laiku.

Iegūtās atziņas noderēs turpmākajos projektos, kuros tiktu uzstāts uz šādu automatizēto testu nepieciešamību vai pat projekta izstrādi, izmantojot *test-driven development* metodi.

```
drcu@drcu-MS-7817: ~
drcu@drcu-MS-7817:~$ ~/node_modules/.bin/mocha -t 10000 ~/teca/unit_tests/tests.js

TECA::Profiles -
  ✓ Change temperature scale to Celsius. (1426ms)
  ✓ Cannot create a profile without a title. (636ms)
  ✓ Cannot create a profile with empty fields. (808ms)
  ✓ Cannot create a profile with invalid task values. (977ms)
  ✓ Validate temperature and time min/max values. (6250ms)
  ✓ Create a profile. (1045ms)
  ✓ Cannot create a profile with a duplicate name. (1722ms)
  ✓ Create a profile using another profile as a template. (1198ms)
  ✓ Edit a profile. (6219ms)
  ✓ Delete a profile. (6241ms)

TECA::Configuration -
  ✓ Current configuration and network interface is being dispalyed. (2597ms)
  ✓ Edit button press - input fields and the right buttons are shown. (917ms)
  ✓ Test configuration field min values. (3004ms)
  ✓ Test configuration field max values. (1741ms)
  ✓ Test all other configuration fields. (600ms)
  ✓ Change the configuration. (4133ms)
  ✓ Cannot edit network interface in DHCP mode. (1029ms)
  ✓ Cannot input invalid address, netmask and gateway. (2419ms)

TECA::Status/Modes -
  ✓ Change temperature scale to Celsius. (1949ms)
  ✓ Correct fields, text and buttons are shown for Protection mode. (7396ms)
  ✓ Correct fields, text and buttons are shown in Stabilization mode. Min and max values validate. (31961ms)
  ✓ Correct fields, text and buttons are shown in Profile mode. (10101ms)
```

Ekrānuzņēmums pēc testu izpildes

4. PROJEKTA ORGANIZĀCIJA

Sistēma tika izstrādāta, lietojot prototipēšanas programmatūras izstrādes dzīves cikla modeli, - paralēli tehnoloģiju apguvei, kas bija nepieciešamas sistēmas izstrādē, tikšanās laikā ar prakses darba devējiem uzņēmumā, tika precizētas un noskaidrotas papildus prasības funkcionalitātei, veiktas izmaiņas jau izstrādātajā sistēmas vienumā.

Lai gan šāda veida iteratīvajai programmatūras izstrādei ir paralēles ar inkrementējošo dzīves cikla modeli, sākotnēji tehnoloģiju un risinājumu nepārzināšanas, kā arī līdz galam nedefinēto prasību dēļ, sistēmas izstrāde tika organizēta pēc evolucionārā prototipēšanas veida horizontālā virzienā, lēnām uzlabojot un pilnveidojot visu izstrādāto funkcionalitāti, pie reizes precizējot prasības un papildinot programmatūras dokumentāciju.

Horizontālā virziena izstrādes dēļ lielākā daļa sistēmas tika pabeigta aptuveni vienā laikā, lai gan funkcionalitāte, kas bija skaidri definēta vai tika pakļauta mazam riskam tikt mainītai, tika pabeigta uzreiz.

Iegūtā pieredze projekta organizēšanā un zināšanas sistēmas izstrādē nākotnē palīdzēs labāk organizēt savu darbu un pieļaut mazāk kļūdu.

Ja būtu uzstāts uz visu prasību definēšanu un precizēšanu jau sākotnēji, tad sistēmas izstrāde noritētu ātrāk un efektīvāk, bez nepieciešamības papildināt vai pārveidot jau izstrādātu funkcionalitāti. Kā zināms, jo sistēma vai modulis ir augstākā pabeigtības stadijā, jo sarežģītāk un dārgāk ir ieviest tajā izmaiņas.

5. KVALITĀTES NODROŠINĀŠANA

5.1. Dokumenta noformēšana

Izstrādājot programmatūras produkta dokumentāciju, tās noformēšanā tika ievēroti valsts standarti programminženierijā:

- LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”;
- LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.

5.2. Programmatūras koda noformēšana

Programmatūras kods viscaur sistēmai ir veidots pēc iespējas līdzīgāks, sekojot turpmākajiem norādījumiem.

5.2.1. Mainīgo nosaukumu veidošana

Visu globālo mainīgo nosaukumi tika veidoti ar lielajiem burtiem, ja iespējams, globālie mainīgie tika loģiski sagrupēti zem vienas datu struktūras, ierobežojot globālo mainīgo daudzumu.

Mainīgo nosaukumi viscaur programmatūras kodā tiek definēti sekojoši – programmatūras pamatfunktionalitātes nodrošināšanai, kas ir rakstīta C valodā, mainīgo nosaukumi un funkciju nosaukumi nepieciešamības gadījumā tiek atdalīti ar apakšsvītru (“_”), bet saskarnes veidošanā (JavaScript programmēšanas valodā un HTML/CSS) tiek izmantots *camelCase* princips, rakstot vārdus kopā, bet, veidojot katru nākamo vārdu aiz pirmā, ar lielo burtu.

Mainīgo nosaukumu veidošanas principi atšķiras, jo pamatfunktionalitātes nodrošināšanai izmantotā JSON formāta apstrādes bibliotēka, izmanto pirmo pieeju, tādēļ, lai nepadarītu kodu grūtāk lasāmu, visa C daļa ir rakstīta pēc tāda principa, bet otra pieeja ir mūsdienīgāka un piemērotāka mājas lapu programmēšanai, jo šādā veidā arī ir definētas JavaScript un jQuery funkcijas, kā arī uzrakstītais kods ir kompaktāks.

Reizēm, lai precīzāk apzīmētu mainīgā saturu, nosaukumā tiek iekļauts kāds aprakstošs lielums, piemēram, saskarnes veidošanā *addProfileBtn*, kur *Btn* apzīmē angļu valodā “*button*” jeb pogu, bet C daļā *fmt_value*, kur *fmt* priedēklis mainīgo izceļ kā JSON bibliotēkas objektu un nošķir no citiem mainīgo tipiem.

Saskarnes programmēšanā funkciju nosaukumi, kas veic AJAX pieprasījumus, tiek veidoti pēc apakšsvītru principa, lai šīs funkcijas acīmredzami nošķirtu no pārējām funkcijām.

Apakšsvītru princips, protams, tiek izmantots arī globālajiem mainīgiem, kuru nosaukumu visur ir ar lielajiem burtiem.

5.2.2. Komentāru rakstīšana

Komentāri ir veidoti divos veidos – blokveida komentāri (*/* komentārs */*) tiek izmantoti, lai paskaidrotu koda loģiku, tajā notiekošo, vienmēr tieši pirms aprakstošā fragmenta (ar izņēmumu vienībtestiem, kur tiek izmantota vienrindes komentēšanas metode, lai nerastos problēmas, gadījumā, ja ir nepieciešams izkomentēt daļu koda, lai palaistu noteiktu daļu testu, bet vienrindas komentāri (*// komentārs*) tiek izmantoti, lai paskaidrotu mainīgo vai kādas īsas darbības nozīmi, ja to ir iespējams izdarīt pāris vārdos.

5.2.3. Koda uzbūve

Pielietojot *DRY* jeb “*don't repeat yourself*” principu, lai izvairītos no koda dublējumiem, viscaur programmatūras kodā, ja viena un tā pati darbība tiek izmantota divās vai vairāk vietās, koda rindiņas tiek nošķirtas no konkrētā moduļa (ja tās nav raksturīgas tikai tam) vai izsaukuma un pārveidotas par funkciju kopējā datnē, kas pēc tam tiek iekļauta visos moduļos, kur nepieciešama.

Funkcijas viscaur pirmkodā ir sarindotas uzreiz pēc to pieminēšanas brīža.

6. KONFIGURĀCIJAS PĀRVALDĪBA

Reizē programmatūras koda rezerves kopija un versiju kontrole tika nodrošināta ar *GIT* versiju kontroles sistēmu, izmantojot *BitBucket* internet vietni, kur tika izveidots repozitorijs.

Izmaiņas tika publicētas visai bieži – gandrīz katru dienu, reizēm vairākas reizes. Vienmēr pēc strādājošiem atjauninājumiem vai svarīgām funkcionalitātes izmaiņām. Iespējams, ka izmaiņu publicēšana ir atlikta uz vēlāku brīdi, ja funkcionalitāte, pie kuras ir bijis strādāts, nav pabeigta līdz galam vai arī izmaiņas ir izjaukušas sistēmas darbību citās vietās, ar mērķi, lai pēc tam būtu vieglāk atklāt kļūdu avotu, salīdzinot pirmkodu.

Programmatūras konfigurācijas pārvaldība ir neatliekama projektu izstrādes sastāvdaļa, bez kuras nav iedomājama pat maza projekta izstrāde. Tā izmantošana var ievērojami samazināt atklādošanas laiku un paaugstināt programmētāju efektivitāti.

7. DARBIETILPĪBAS NOVĒRTĒJUMS

Bez iepriekšējas pieredzes šādu vai pat cita veidu sistēmas izstrādē novērtēt izstrādājama programmaprodukta darbietilpību ir ļoti sarežģīti, ja ne neiespējami. Šī iemesla dēļ, pirms darba uzsākšanas pie projekta, tika uzklausīts prakses darba devēja kā eksperta slēdziens par to, ka šādu sistēmu ir iespējams izstrādāt norādītās prakses ietvaros ar darbietilpību vismaz 3 personmēnešiem.

Sistēmas darbietilpības novērtējumam tika izmantota “*Basic COCOMO*” darbietilpības novērtēšanas metode ar formulu $A*(KLOC/1000)^B$ (*KLOC* ir programmrindiņu skaits tūkstošos), kurai tika izmantotas konstantes $A=3.0$ un $B=1.12$, kas raksturo, ka izstrādē piedalās “vidēja izmēra komanda ar dažādu pieredzes līmeni, strādājot ar stingrām un ne pārāk stingrām prasībām”.

Tika izraudzīts šis konstanšu piedāvātais variants, ņemot vērā, ka pirms darba uzsākšanas sistēmas izstrādātājam šāda pieredze nebija, kā arī pieminēto *stingro prasību* dēļ, kas attiecas uz *Raspberry Pi* ierobežoto operatīvās atmiņas daudzumu, ierobežotajiem skaitļošanas resursiem un izstrādājamās sistēmas dabu – tā ir paredzēta ražošanas nolūkiem, tātad tai ir jāstrādā netraucēti nepārtrauktā režīmā.

Formulai pielietojot patstāvīgi izstrādāto C, JavaScript, HTML (atkārtoti neskaitot *.*shtml* datnes), CSS un Bash (neskaitot *makefile* datnes) programmkoda rindiņu skaitu rezultāts sanāca $3.0*(8448/1000)^{1.12} = 32.7$ personmēneši. Ir redzams, ka vismaz šajā gadījumā pielietot COCOMO metodi, lai noteiktu darbietilpību, nebūtu adekvāti.

Tā kā paralēli praksē pavadītajam laikam, kad tika izstrādāta sistēma, tika skaitīts sistēmas izstrādei veltīto stundu skaits (ko mēdz arī uzņēmos izmantot par darbietilpības atspoguļojumu), ir iespējams objektīvi pārliecināties par sistēmas darbietilpību.

Realitātē nepieciešamo lietu apguvei, uzstādīšanai un sistēmas izstrādei, personstundas pārvēršot personmēnešos (tās dalot ar 176), tika patērēti aptuveni **3.8** personmēneši.

Zemāk esošajās tabulās ir attēlots programmrindiņu sadalījums pa programmēšanas valodām un datnēm.

7.1. C (pamatfunkcionalitāte)

Pirmkoda datne	Aptuvenais rindiņu skaits
defines.h	176
conf.c	210
get_logs.c	29
get_status.c	25
get_sysinfo.c	29
gpio.h	64
gpio.c	132
i2c.h	135
i2c.c	196
includes.c	255
interfaces.c	329
main.c	1245
modes.c	214
profiles.c	244
Kopā	3283

7.2. JavaScript (saskarnes un vienībtesti)

Pirmkoda datne	Aptuvenais rindiņu skaits
conf.inc	328
logs.inc	362
sysinfo.inc	219
modes.inc	275
profiles.inc	445
status.inc	415
js.js	100
main/js.js	118
rpi/js.js	653
tests.js	1183
Kopā	4098

7.3. HTML (saskarnes)

Pirkoda datne	Aptuvenais rindiņu skaits
index.shtml, conf.shtml, logs.shtml, modes.shtml, profiles.shtml, sysinfo.shtml	27
conf.inc	101
logs.inc	55
sysinfo.inc	44
modes.inc	36
index_local.html	180
profiles.inc	56
status.inc	67
Kopā	566

7.4. CSS (saskarnes)

Pirkoda datne	Aptuvenais rindiņu skaits
rpi/style.css	106
main/style.css	240
Kopā	346

7.5. Bash

Pirkoda datne	Aptuvenais rindiņu skaits
daemon.sh	14
setup.sh	96
update.sh	45
Kopā	155

8. PROGRAMMATŪRAS KODA FRAGMENTI

i2c.h

```
/*
 * @brief Write "00001001" (0x09) to reset settings ("0000_100x" is default).
 *
 * IMPORTANT: Must be called to read internal temperature if function
 "activate_thermistor" has been previously called.
 *
 * @return 0 on success, negative error code otherwise.
 */
int activate_internal();
```

main.c

```
/* Adjust the temperature according to the active mode and report temperature related
anomalies. */
void adjust_temperature() {

    /* Protection mode. The heater got switched off already in the
    rpc_method_start_protection call. */
    if(STATUS.mode == MODES.protection) {
        /* Bring down the temperature. */
        if(STATUS.temperature >= ADJUSTMENT.first_high) {
            if(!ADJUSTMENT.adjusted) ADJUSTMENT.adjusted = 1; // In protection let it
            always fall through the first check (TC/devices stopped working, might not need to
            cool).

                else if(STATUS.temperature >= ADJUSTMENT.second_high) cooler_on();

            /* Normalized temperature. */
        } else {
            cooler_off();
            ADJUSTMENT.adjusted = 0;
            ADJUSTMENT.target_temperature_reached = 1;
        }

        /* Report error if it's not done already. */
        if(!LOGGED_TEMPERATURE_FAIL) {

            /* Temperature too high. */
            if(STATUS.temperature > DELTA_OFF) {
                LOGGED_TEMPERATURE_FAIL = 1;
                snprintf(LOG_MSG, MSG_LEN, "Temperature %.1f C degrees over 35 C, %.1f C
                degrees over allowable temperature.", STATUS.temperature - 35, STATUS.temperature -
                DELTA_OFF);
                log_information(LOGGING_LEVEL.warning, "S", LOG_MSG);
            }
        }
    }

    /* Stabilization or profile mode. */
    else {
```

```

    /* Correct the temperature. */
    if(STATUS.temperature <= ADJUSTMENT.first_low || STATUS.temperature >=
ADJUSTMENT.first_high) {
        if(STATUS.temperature <= ADJUSTMENT.first_low) {
            if(!ADJUSTMENT.adjusted) {
                ADJUSTMENT.adjusted = 1;
                cooler_off();
            } else if(STATUS.temperature <= ADJUSTMENT.second_low) {
                heater_on();
            }
        }

        if(STATUS.temperature >= ADJUSTMENT.first_high) {
            if(!ADJUSTMENT.adjusted) {
                ADJUSTMENT.adjusted = 1;
                heater_off();
            } else if(STATUS.temperature >= ADJUSTMENT.second_high) {
                cooler_on();
            }

            /* Alert the user if stabilization temperature is not reached in the
STABILIZATION_COOLING_WAIT_TIME (specified in defines.h) time. */
            if(STATUS.mode == MODES.stabilization && ( (SYSINFO.uptime -
STABILIZATION_STARTED) > STABILIZATION_COOLING_WAIT_TIME ) &&
!LOGGED_STABILIZATION_FAIL) {
                LOGGED_STABILIZATION_FAIL = 1;
                snprintf(LOG_MSG, MSG_LEN, "Failed to reach stabilization temperature of
%.1f C after %.1f hours.", STATUS.target_temperature,
(float)STABILIZATION_COOLING_WAIT_TIME/3600);
                log_information(LOGGING_LEVEL.warning, STATUS.profile, LOG_MSG);
            }
        }
    }

    /* Temperature is ok. */
} else {
    cooler_off();
    heater_off();
    ADJUSTMENT.adjusted = 0;
    ADJUSTMENT.target_temperature_reached = 1;
}

/* TASK_SUCCESSFUL - temperature profile has failed if the temperature
goes out of the allowable range (TEMP - DELTA_ON < TEMP < TEMP + DELTA_OFF).
If target temperature has reached, then have to check if it's in the allowable
range.
*/
if(ADJUSTMENT.target_temperature_reached) {

    /* Report error if it's not done already. */
    if(!LOGGED_TEMPERATURE_FAIL) {

        /* Temperature too high. */
        if(STATUS.temperature > DELTA_OFF) {
            LOGGED_TEMPERATURE_FAIL = 1;

```


9. NOBEIGUMS

Prakses ietvaros kā kvalifikācijas darbs tika veiksmīgi izstrādāta uzņēmuma “SAF Tehnika” termoskapju vadības sistēma, kas tiek izmantota ražošanas vajadzībām, testējot izstrādājamās un izstrādātās iekārtas datu pārraidei.

Kvalifikācijas darba autoram iepriekš nebija pieredze šādu sistēmu izstrādē, tādēļ prakses laikā tika pilnveidotas esošās iemaņas saskarņu izstrādē un tehnoloģijās (*JavaScript, AJAX*), tika iegūtas vērtīgas zināšanas reāllaika sistēmu izstrādē ar vairākiem vienlaicīgiem procesiem, zema līmeņa programmēšanu C programmēšanas valodā, *Linux* operētājsistēmu, *RPC* u.c. tehnoloģijām.

Iegūtās zināšanas un uzkrātā pieredze sistēmas izstrādes laikā noteikti noderēs turpmākajos projektos.

Kvalifikācijas darbs „*Uzņēmuma “SAF Tehnika” termoskapju vadības sistēma*”
izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie
informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Druvis Cukurs* _____ .06.2015.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *Dr.dat. Dainis Dosbergs* _____ .06.2015.

Recenzents: *M.dat. Agnis Škuškovniks*

Darbs iesniegts 01.06.2015.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2015. prot. Nr. _____

Komisijas sekretārs(-e): _____