



LATVIJAS
UNIVERSITĀTE
ANNO 1919

DATORIKAS FAKULTĀTE

“GUDRĀ MĀJA” LIETOTĀJA SASKARNE

KVALIFIKĀCIJAS DARBS

Autors: Gustavs Helmutis Felsbergs

Studenta apliecības Nr. gf13003

Darba vadītājs: Dipl. dat. Guntars Šmitiņš

Rīga 2016

ANOTĀCIJA

Kvalifikācijas darbā izstrādāta lietotnes “Gudrā Māja” lietotāja saskarne, kas lietotājam nodrošina visu lietotnē paredzēto funkciju pilnvērtīgu izmantošanu. Izstrādājamā programmaprodukta mērķis ir nodrošināt lietotājiem attālinātu privātīpašuma ierīču kontroli un sensoru mērījumu apskati, izmantojot kopējās sistēmas REST tīmekļa servissus. Lietotāja saskarne nodrošina ērtu lietotnes funkciju izmantošanu gan mobilajās ierīcēs, gan datoros. Lietotāja saskarnes izstrādei izmantots SAPUI5 ietvars, JavaScript programmēšanas valoda, jQuery bibliotēka un JSON formāts.

Lietotni ir paredzēts izmantot privātīpašumu iedzīvotājiem, kas sistēmā ir reģistrēti lietotāji, kam ir pieejamas tikai noteiktas funkcijas, un īpašuma pārvaldniekiem, kas sistēmā ir administratori, kam ir pieejamas visas funkcijas.

Atslēgas vārdi: SAPUI5, JavaScript, JSON, REST tīmekļa serviss.

ABSTRACT

The main goal was to develop front-end for the application “Smart House” allowing usage of this application’s functions for its user. The goal of this software product is to provide users with remote control of devices in their house and view measurements of sensors. Data used by front-end are provided by using systems’ REST web services. Responsive design used in UI development allows usage of the application on computers as well as on mobile devices. User Interface was developed by using SAPUI5 framework, JavaScript programming language, jQuery library and JSON format

Usage of the application is restricted by two roles: regular users and administrators. Residents of a private property are regular users in the system and have only certain application’s functions available, and managers of the private property are administrators in the system and have all the application’s functions available to them.

Keywords: SAPUI5, JavaScript, JSON, REST web service.

SATURA RĀDĪTĀJS

ANOTĀCIJA	2
ABSTRACT	3
IEVADS	9
2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	11
2.1 Ievads	11
2.1.1 Nolūks	11
2.1.2 Darbības sfēra.....	11
2.1.3 Saistība ar citiem dokumentiem	12
2.1.4 Dokumenta pārskats	12
2.2 Vispārējais apraksts	12
2.2.1 Produkta perspektīva	12
2.2.2 Produkta funkcijas.....	13
2.2.3 Lietotāja raksturiezīmes	13
2.2.4 Vispārēji ierobežojumi	14
2.3 Funkcionālās prasības	14
2.3.1. Lietotāja reģistrācija.....	14
2.3.2. Lietotāja autentifikācija.....	15
2.3.3. Lietotāja dzēšana	16
2.3.4. Lietotāja rediģēšana.....	18
2.3.4. Apkures sistēmas trīszaru vārstu temperatūras uzstādīšana	19
2.3.6. Apkures sistēmas izslēgšana	20
2.3.7 Ventilācijas regulēšana.....	21
2.3.8. Vārtu kontrole	22
2.3.9. Vārtu apturēšana.....	23

2.3.10. Drošības zonu kontrole.....	24
2.3.11. Drošības sistēmas kontrole.....	25
2.3.12. Īpašuma gaismas slēdžu kontrole.....	26
2.3.13. Visu gaismas slēdžu kopējā kontrole.....	27
2.3.14. Sensoru mērījumu pārskata attēlošana.....	28
2.3.15. Sensoru izvērstās izvēlnes filtrēšana.....	29
2.3.16. Valodas maiņa un uzstādīšana.....	30
2.3.17. Ventilācijas CO2 līmeņa uzstādīšana.....	31
2.4. Nefunkcionālās prasības.....	32
2.4.1. Veiktspējas prasības.....	32
2.4.2. Drošības prasības.....	32
2.4.3. Pieejamība.....	32
2.4.4. Uzturamība.....	32
2.5. Ārējās saskarnes prasības.....	32
3. POGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	34
3.1. Ievads.....	34
3.1.1. Nolūks.....	34
3.1.2. Darbības sfēra.....	34
3.1.3. Saistība ar citiem dokumentiem.....	34
3.2. Trasējamības tabula.....	35
3.3. Moduļu dekompozīcija.....	37
3.3.1. Administrator modulis.....	37
3.3.2. Session modulis.....	38
3.3.3. Heating modulis.....	38
3.3.4. Ventilation modulis.....	39
3.3.5. Gates modulis.....	39

3.3.6. Security modulis	40
3.3.7. Light modulis.....	40
3.3.8. Overview modulis.....	41
3.3.9. Language modulis	41
3.3. Atkarības apraksts.....	42
3.3.1. Pirmā līmeņa datu plūsmu diagramma.....	42
3.3. Ārējā saskarne.....	44
3.3.1. Ārējās saskarnes pielāgošanās ierīcei un ekrāna izmēram	44
3.3.1. Valodas izvēle	45
3.3.2. Lietotnes vispārējie kļūdu paziņojumi	46
3.3.2. Autorizācijas skats.....	46
3.3.3. Funkciju izvēles skats.....	47
3.3.4. Mājas kopējā stāvokļa skats	48
3.3.5. Ventilācijas skats.....	50
3.3.6. Vārtu kontroles skats.....	51
3.3.7. Apkures sistēmas kontroles skats.....	52
3.3.8. Drošības sistēmas kontroles skats	53
3.3.9. Gaismas kontroles skats	55
3.3.10. Pārskata skats	56
3.3.11. Administratora skats.....	58
3.3.12. IP Kameru skats	60
4. TESTĒŠANAS DOKUMENTĀCIJA.....	61
4.1. Ievads	61
4.2. Testēšanas žurnāls.....	61
4.2.2. Lietotāja autentifikācija.....	62
4.2.3. Lietotāja dzēšana	63

10.3.4. “generateFilter”	86
SECINĀJUMI	88
IZMANTOTĀ LITERATŪRA	89
PIELIKUMS.....	90

IEVADS

“Gudrā Māja” ir lietotne, kas sniedz papildus ērtības tās lietotājiem, nodrošinot iespēju attālināti kontrolēt savā īpašumā esošās ierīces nodrošina papildus ērtības un drošību, dodot iespēju īpašumu iedzīvotājiem attālināti kontrolēt īpašumā esošās ierīces, izmantojot datoru vai viedtālruni ar interneta pieslēgumu. Lietotne sniedz papildus drošību, ļaujot lietotājam attālināti ieslēgt un izslēgt īpašumā signalizāciju, kā arī ļaujot redzēt īpašuma sensoru mērījumus, parādot, vai īpašuma stāvoklis atbilst lietotāja vēlmēm. Šajā darbā tiek izstrādāta lietotāja saskarne lietotnei “Gudrā Māja”.

Izstrādājamā programmaprodukta – “Gudrā Māja” lietotāja saskarnes – mērķis ir ērti pārskatāmā veidā attēlot lietotājam no tīmekļa servisiem iegūtos datus, izmantojot datu piesaisti, kā arī nodrošināt lietotnes funkciju izmantošanu, ļaujot lietotājam ērti rediģēt datu bāzē esošo informāciju, izmantojot tīmekļa servisu. Lietotāja saskarne ļauj izmantot lietotni “Gudrā Māja” jebkuram, kurš māk izmantot datoru vai viedtālruni, jo pārslēgšanās starp funkcijām un to izmantošana ir vienkārša un saprotama.

PAMATDEFINĪCIJAS

PPS - Programmatūras prasību specifikācija

PPA – Programmatūras projektējuma apraksts

SAPUI5 – SAP user interface for HTML5: bibliotēku apkopojums, kas paredzēts tīmekļa lietotņu veidošanai.

JSON – JavaScript Object Notation: datu apmaiņas formāts.

MVC – “Model View Controller” arhitektūras šablons.

Datu piesaiste - Modeļa datu piesaistīšana elementam, nodrošinot automātisku datu atspoguļošanu.

REST – tīmekļa servisi, kas nodrošina ērtu datu apmaiņu starp lietotāja saskarni un serveri.

AJAX – metode datu apmaiņai ar serveri, atjaunojot tīmekļa lapas daļas, nepārlādējot visu lapu.

ID – unikāls identifikācijas numurs

Sensori – mērpārveidotājs vai ierīce, kas tieši uztver mērījamo lielumu (piemēram, spiedienu, temperatūru, deformāciju, spriegumu, gaismas daudzumu, vielas stāvokli) un veido tā vērtībām atbilstošu signālu virkni.

Administrators - reģistrēts lietotājs, kuram piešķirtas lietotnes “Gudrā māja” pilnas pārvaldes tiesības.

Jenkins - nepārtrauktās integrācijas rīks, kas ļauj automātiski veikt būvējumus pēc izstrādātāju veiktajām koda izmaiņām.

Atlassian JIRA – sistēma, kas paredzēta projekta uzdevumu un problēmu pieteikumu apstrādei.

SonarQube - rīks koda kvalitātes pārvaldīšanai.

Startrack – rīks, ar kura palīdzību uzskaita darbu un tam veltīto laiku.

SVN – versiju kontroles rīks

2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

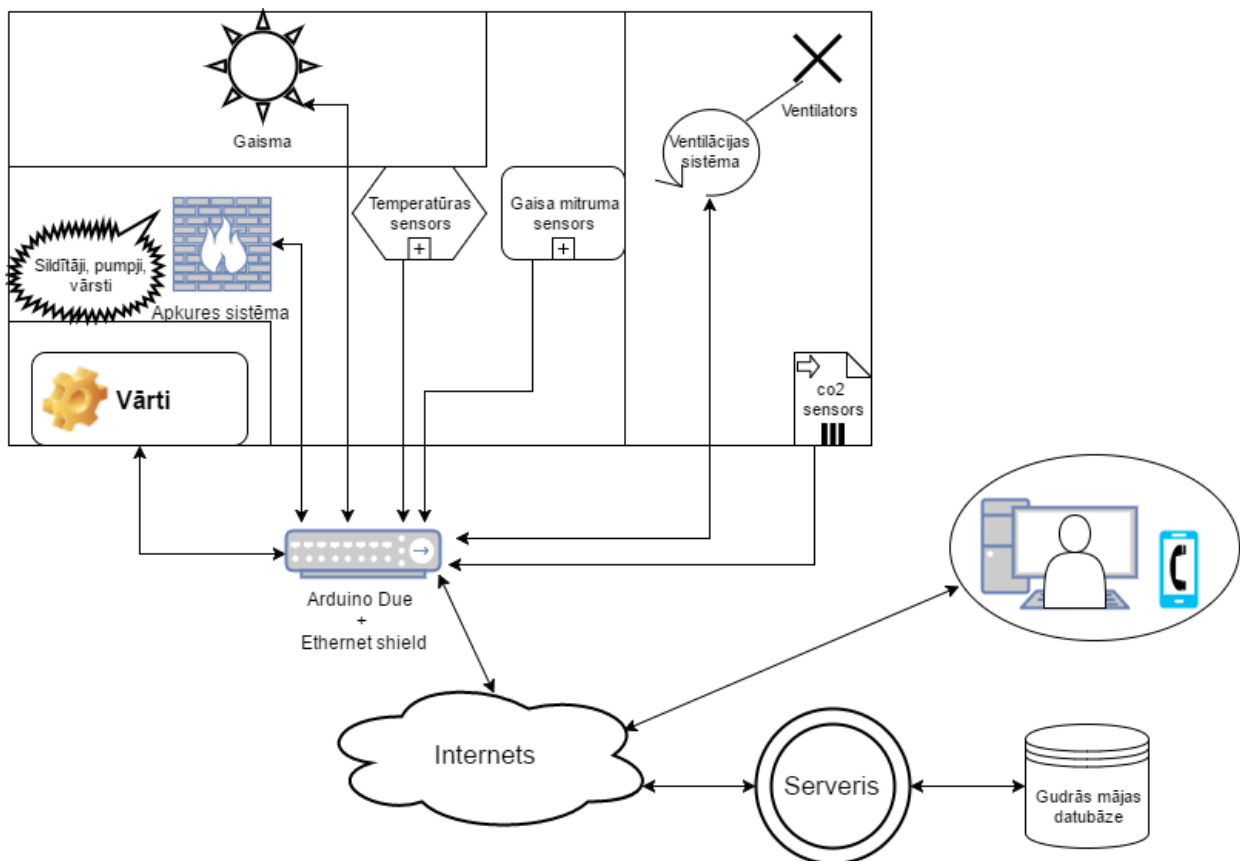
2.1 Ievads

2.1.1 Nolūks

Šīs programmatūras prasību specififikācijas nolūks ir vienozīmīgi un nepārprotami formulēt lietotnes “Gudrā Māja” lietotāja saskarnes, funkcionālās, nefunkcionālās prasības un vispārējo aprakstu, tādējādi nodrošinot kvalitatīvas lietotāja saskarnes izstrādi.

2.1.2 Darbības sfēra

Izstrādājamais programms produkts ir lietotnes “Gudrā Māja” lietotāja saskarne. Šī produkta mērķis ir, izmantojot kopējās sistēmas REST tīmekļa servisu, ļaut ērtā veidā attālināti kontrolēt īpašumā esošas ierīces, redzēt īpašuma sensoru mērījumus un ierīču stāvokļus. (skat. Attēls 1) Izmantojot šo lietotni, privātīpašuma iedzīvotājiem un īpašniekiem nav jābūt klātienē, lai redzētu un kontrolētu īpašumā esošo situāciju.



Attēls 1

2.1.3 Saistība ar citiem dokumentiem

PPS izveidota, ievērojot standarta LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis” prasības.

2.1.4 Dokumenta pārskats

Dokuments sastāv no šādām daļām:

- Ievads, kas satur dokumenta nolūku un īsi apraksta programmaproduktu.
- Vispārējais apraksts, kas satur īsu informāciju par programmaprodukta perspektīvu, funkcijām, lietotāja raksturiezīmēm un vispārējiem ierobežojumiem
- Funkcionālās prasības
- Nefunkcionālās prasības

2.2 Vispārējais apraksts

2.2.1 Produkta perspektīva

Lietotne “Gudrā Māja” ir paredzēta lietošanai privātpašumu iedzīvotājiem un pārvaldniekiem. Izmantojot šo lietotni, īpašuma pārvaldnieks un iedzīvotāji var attālināti pārvaldīt īpašumā esošās ierīces un redzēt sensoru mērījumus, kas ļauj pārliecināties, ka īpašuma stāvoklis atbilst lietotāja vēlmēm.

Lietotne tiek veidota kā tīmekļa vietne, kas nodrošina ērtu lietošanu gan datorā, gan mobilajā ierīcē. Lietotne izmanto Arduino mikrokontrolieri, kam pieslēgti sensori un ierīces

(skat. piem. pielikumā Attēls 33). Attīstoties lietu internetam un šīm ierīcēm un sensoriem kļūstot lētākiem, šādas sistēmas varētu lietot liela daļa pasaules iedzīvotāju.

2.2.2 Produkta funkcijas

Programmproduktam ir paredzēta šāda funkcionalitāte:

Reģistrētiem lietotājiem:

- Lietotāja autentifikācija;
- Apkures sistēmas trīszaru vārtu temperatūras regulēšana un apkures sistēmas izslēgšana;
- Ventilācijas regulēšana;
- Vārtu kontrole un apturēšana;
- Drošības zonu un sistēmas kontrole;
- Īpašuma gaismas avotu kontrole;
- Sensoru filtrēšana un merījumu apskate;
- Valodas maiņa un uzstādīšana;

Administratoriem paredzēta iepriekš aprakstītā funkcionalitāte un papildus funkcionalitāte:

- Lietotāju reģistrēšana;
- Lietotāju rediģēšana;
- Lietotāju dzēšana;

2.2.3 Lietotāja raksturiezīmes

Sistēmu paredzēts lietot privātīpašuma iedzīvotājiem un īpašniekiem. Pieejamo funkcionalitāti nosaka trīs lomas:

Administratori - privātīpašuma īpašnieki/pārvaldnieki, kas papildus reģistrēto lietotāju funkcionalitātei var pievienot lietotājus, dzēst lietotājus un rediģēt lietotājus.

Reģistrēti lietotāji - privātīpašuma iedzīvotāji, kas var aplūkot datus un attālināti kontrolēt pieejamās ierīces.

Sistēmas konfigurators - uzstāda lietotni konkrētai mājai - kontrolieru pieslēgšana sistēmai, jaunas informācijas ievadīšana, pārējās mājas uzbūves definēšana datubāzē. Sistēmas konfiguratora lietotāja saskarnes izstrāde neietilpst šajā kvalifikācijas darbā.

2.2.4 Vispārēji ierobežojumi

- Sistēma atbalsta Google Chrome un Mozilla Firefox pēdējās versijas un Internet Explorer 11+ pārlūkprogrammas.
- Lietotāja saskarnes izveide jāveic, izmantojot SAPUI5 ietvaru.
- Lietotāja saskarnei ir jāatbalsta tekstu valodas maiņa.
- Sistēmai drīkst piekļūt tikai autorizēti lietotāji.
- Lietotāja saskarnei jābūt lietojamai mobilajā tālrunī un datorā.

2.3 Funkcionālās prasības

2.3.1. Lietotāja reģistrācija

Identificējums – FP1

Ievads
Funkcija ir pieejama tikai administratoriem, tās mērķis ir pievienot jaunu lietotāju.
Ievades dati
<ul style="list-style-type: none">• Lietotājvārds – teksts, kas nedrīkst pārsniegt 25 simbolus un nedrīkst sastāvēt no mazāk par 4 simboliem.• Lietotāja vārds – teksts, kas nedrīkst pārsniegt 50 simbolus un nedrīkst sastāvēt no mazāk kā 2 simboliem.• Lietotāja uzvārds – teksts, kas nedrīkst pārsniegt 50 simbolus un nedrīkst sastāvēt no mazāk kā 2 simboliem.

<ul style="list-style-type: none"> • Lietotāja e-pasts - teksts, kas nedrīkst pārsniegt 50 simbolus un nedrīkst sastāvēt no mazāk par 6 simboliem. • Lietotāja parole – teksts, kas nedrīkst pārsniegt 20 simbolus un nedrīkst sastāvēt no mazāk par 6 simboliem. • Pieprasījums reģistrēt jaunu lietotāju. • Visi lauki ir obligāti aizpildāmi.
Apstrāde
<ol style="list-style-type: none"> 1. Tiek izveidots JSON formāta datu fragments ar ievadīto lauku datiem. 2. Izmantojot rest/users/user/{house_id} tīmekļa servisu ar POST metodi, dati tiek nosūtīti lietotnes aizmugursistēmai, kur tiek izveidots jauns lietotājs.
Izvade
<ol style="list-style-type: none"> 1. Uz ekrāna parādās dialoglodziņš ar paziņojumu par veiksmīgu reģistrāciju. 2. Tiek parādīti lietotāji, ieskaitot jauno izveidoto lietotāju.
Kļūdu paziņojumi
<ul style="list-style-type: none"> • Kļūdas paziņojums par neaizpildītu obligātu/kļūdaini aizpildītu lauku(iem) un iezīmēti neaizpildītie/kļūdainie lauki. • Kļūdas paziņojums par eksistējošu lietotāju, kura lietotājvārds ir tāds pats kā izveidojamajam lietotājam.

2.3.2. Lietotāja autentifikācija

Identificējums – FP2

Ievads
Funkcijas mērķis ir ļaut lietotājam pieteikties sistēmā.
Ievades dati
<ul style="list-style-type: none"> • Lietotājvārds; • Lietotāja parole; • Visi lauki ir obligāti aizpildāmi.

Apstrāde
<ol style="list-style-type: none"> 1. Ja visi lauki ir pareizi aizpildīti, tiek iegūtas lauku vērtības un izveidots JSON datu fragments. 2. Dati tiek sūtīti serverim, kur tiek pārbaudīts, vai eksistē lietotājs ar ievadīto lietotājvārdu un paroli.
Izvade
<ol style="list-style-type: none"> 1. Lietotājs tiek pieslēgts sistēmai. 2. Parādās galvenais skats, kur lietotājs var izvēlēties kādu no sev pieejamajām funkcijām.
Kļūdu paziņojumi
<ul style="list-style-type: none"> • Ja ievadītā lietotājvārda un paroles kombinācija neeksistē datu bāzē, tiek parādīts ziņojums, ka lietotājvārds vai parole nav pareiza.

2.3.3. Lietotāja dzēšana

Identificējums – FP3

Ievads
Funkcijas mērķis ir dzēst lietotāju. Šī funkcija ir pieejama tikai administratoriem.
Ievades dati
Pieprasījums un apstiprinājums dzēst lietotāju.
Apstrāde
<ol style="list-style-type: none"> 1. Izmantojot datu piesaisti, tiek iegūts lietotāja lietotājvārds. 2. Izmantojot rest/users/user/{username} tīmekļa servisu ar DELETE metodi, dati tiek sūtīti serverim, kur tiek pārbaudīts, vai pieprasījumu izdara administrators, un tiek izdzēsts lietotājs.

Izvade
<ol style="list-style-type: none">1. Paziņojums par veiksmīgu lietotāja dzēšanu.2. Lietotāju tabula pārlādēta bez izdzēstā lietotāja.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

2.3.4. Lietotāja rediģēšana

Identificējums – FP4

Ievads
Funkcijas mērķis ir rediģēt lietotāja informāciju.
Ievades dati
<ul style="list-style-type: none">• Jaunais lietotāja vārds – teksts, kas nedrīkst pārsniegt 50 simbolus un nedrīkst sastāvēt no mazāk kā 2 simboliem.• Jaunais lietotāja uzvārds – teksts, kas nedrīkst pārsniegt 50 simbolus un nedrīkst sastāvēt no mazāk kā 2 simboliem.• Jaunais lietotāja e-pasts - teksts, kas nedrīkst pārsniegt 50 simbolus un nedrīkst sastāvēt no mazāk par 6 simboliem.• Jaunā lietotāja parole – teksts, kas nedrīkst pārsniegt 20 simbolus un nedrīkst sastāvēt no mazāk par 6 simboliem.• Pieprasījums rediģēt informāciju.
Apstrāde
<ol style="list-style-type: none">1. Tiek izveidots JSON formāta datu fragments ar ievadīto lauku datiem.2. Izmantojot rest/users/user/{username} tīmekļa servisu ar PUT metodi, dati tiek nosūtīti lietotnes aizmugursistēmai, kur tiek rediģēta lietotāja informācija.
Izvade
<ol style="list-style-type: none">1. Uz ekrāna parādās dialoglodziņš ar paziņojumu par veiksmīgu lietotāja informācijas rediģēšanu.2. Lietotāja vecās informācijas vietā tiek uzrādīta jaunā informācija.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Kļūdas paziņojums par neaizpildītu obligātu/kļūdaini aizpildītu lauku un iezīmēts neaizpildītais/kļūdainais lauks.

- Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

2.3.4. Apkures sistēmas trīszaru vārstu temperatūras uzstādīšana

Identificējums – FP5

Ievads
Funkcijas mērķis ir ļaut lietotājam uzstādīt trīszaru vārstu temperatūru, tādējādi regulējot apkures sistēmu. Ja kāda no trīszaru vārstu temperatūrām ir uzstādīta mazāka, nekā attiecīgā trīszaru vārsta faktiskā temperatūra, tā karsēšana tiek pārtraukta.
Ievades dati
<ul style="list-style-type: none"> • Temperatūra, ko lietotājs vēlas izstādīt izvēlētajam trīszaru vārstam, temperatūrai jābūt robežās no 0°C līdz 75°C. • Pieprasījums nomainīt trīszaru vārsta temperatūru.
Apstrāde
<ol style="list-style-type: none"> 1. Pārbaude, vai ievadītā vērtība ir robežās starp 0 un 75°C. 2. Izmantojot datu piesaisti, tiek iegūts trīszaru vārsta ID un ievadītā uzstādāmā temperatūra. 3. Iegūtie dati tiek nosūtīti serverim, izmantojot rest/heatingsystem/heatingsystem/toreach servisu ar PUT metodi, kur tiek uzstādīta ievadītā temperatūra.
Izvade
<ol style="list-style-type: none"> 1. Kad tiek sagaidīti atbildes dati no lietotnes aizmugursistēmas, kas satur attiecīgā trīszaru vārsta ID un jauno uzstādīto temperatūru, šie dati tiek piesaistīti trīszaru vārstam, kam veiktas izmaiņas.

Kļūdu paziņojumi
<ul style="list-style-type: none"> • Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies. • Ja tiek ievadīta temperatūra, kas ir zem 0°C vai virs 75°C, ir decimāls skaitlis vai nav skaitlis, tiek parādīts dialoglodziņš, kas paziņo, ka ievadītā temperatūra ir nepareiza. Šī validācija notiek gan klienta pusē, gan server pusē.

2.3.6. Apkures sistēmas izslēgšana

Identificējums – FP6

Ievads
Funkcijas mērķis ir ļaut lietotājam izslēgt apkures sistēmu. Ja kāda no trīszaru vārstu temperatūrām ir uzstādīta mazāka, nekā attiecīgā trīszaru vārsta faktiskā temperatūra, tā karsēšana tiek pārtraukta.
Ievades dati
<ul style="list-style-type: none"> • Pieprasījums izslēgt apkures sistēmu - pieprasījumu iespējams veikt tikai pie nosacījuma, ka apkures sistēma nav izslēgta jeb vismaz viens trīszaru vārsts ir uzstādīts uz temperatūru lielāku par 0°C.
Apstrāde
<ol style="list-style-type: none"> 1. Tiek izsaukts rest/heatingssystem/heatingssystem/{house_id}/off tīmekļa serviss ar GET metodi. 2. Trīszaru vārstiem tiek piesaistīts saņemtais JSON fails.
Izvade
<ol style="list-style-type: none"> 1. Visu trīszaru uzstādītās temperatūras ir 0°C.
Kļūdu paziņojumi
<ul style="list-style-type: none"> • Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

2.3.7 Ventilācijas regulēšana

Identificējums – FP7

Ievads
Šīs funkcijas mērķis ir nodrošināt lietotājam iespēju redzēt ventilatoru stāvokli un ventilācijas sensoru mērījumus, kā arī mainīt ventilatoru stāvokli.
Ievades dati
Lietotājs iestata nepieciešamo ventilatora stāvokli.
Apstrāde
<ol style="list-style-type: none">1. Izmantojot datu piesaisti un lietotāja ievadi, tiek iegūts ventilatora ID un uzstādāmais režīms.2. Izmantojot rest/ventilationsystem/ventilator/powerstate tīmekļa servisu ar PUT metodi, serverim tiek nosūtīti iegūtie dati JSON formātā, serverī tiek nomainīts ventilatora režīms.
Izvade
<ol style="list-style-type: none">1. No servera saņemtie dati tiek attēloti uz ekrāna kā uzstādītais režīms atbilstošajam ventilatoram.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

2.3.8. Vārdu kontrole

Identificējums – FP8

Ievads
Šīs funkcijas mērķis ir nodrošināt attālinātu īpašuma vārdu kontroli.
Ievades dati
<ol style="list-style-type: none">1. Pieprasījums atvērt vārtus – pieprasījumu iespējams veikt tikai pie nosacījuma, ka vārti ir aizvērti vai apturēti;2. Pieprasījums aizvērt vārtus – pieprasījumu iespējams veikt tikai pie nosacījuma, ka vārti ir atvērti vai apturēti;
Apstrāde
Pieprasot vārdu atvēršanu vai aizvēršanu: <ol style="list-style-type: none">1. Tiek iespējota vārdu apturēšana;2. Cikliski tiek atjaunināts vārdu stāvoklis, izmantojot rest/gates/gate/all/{house_id} tīmekļa servisu ar GET metodi, lai lietotājs varētu zināt, vai vārti atrodas starpstāvoklī (atvēršanas/aizvēršanas procesā);3. Izmantojot datu piesaisti, tiek iegūts vārdu ID un pieprasījuma veids (atvērt/aizvērt);4. Dati tiek sūtīti serverim, izmantojot rest/gates/gate/state servisu.
Izvade
<ol style="list-style-type: none">1. Ja pieprasījums vārdu aizvēršanai vai atvēršanai veiksmīgi saņemts aizmugursistēmā, automātiski atjaunojoties vārdu stāvoklim, parādās attiecīgais starpstāvoklis, kas glabājas datu bāzē. Kad vārti nonākuši galējā stāvoklī, no aizmugursistēmas tiek saņemti dati JSON formātā par vārtiem, kam pieprasījums tika veikts, un šie dati tiek ievietoti attiecīgo vārdu tabulas rindā.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

2.3.9. Vārtu apturēšana

Identificējums – FP9

Ievads
Funkcijas mērķis ir ļaut ārkārtas gadījumā apturēt vārtus, kas atrodas kustībā.
Ievades dati
1. Pieprasījums apturēt vārtus, ko iespējams veikt tikai pie nosacījuma, ka vārti atrodas starpstāvoklī (atvēršanas vai aizvēršanas procesā).
Apstrāde
1. Tiek iegūts vārtu ID, izmantojot datu piesaisti. 2. Tiek izsaukts rest/gates/gate/state tīmekļa serviss, ar kuru serverim nosūta vārtu ID un STOP stāvokli.
Izvade
1. Tiek ielādēti atbildes dati par atbilstošajiem vārtiem. 2. Tiek parādīts jaunais vārtu stāvoklis (apturēti).
Kļūdu paziņojumi
<ul style="list-style-type: none">Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

2.3.10. Drošības zonu kontrole

Identificējums – FP10

Ievads
Šīs funkcijas mērķis ir nodrošināt lietotājam īpašuma zonu signalizācijas ieslēgšanu un izslēgšanu, drošības zonu stāvokļu attēlošanu.
Ievades dati
<ul style="list-style-type: none">• Pieprasījums ieslēgt/izslēgt signalizāciju īpašuma drošības zonā;
Apstrāde
<ol style="list-style-type: none">1. Izmantojot datu piesaisti, tiek iegūts īpašuma zonas ID.2. Izmantojot rest/securitysystem/securityzone/mode tīmekļa servisu ar PUT metodi, serverim tiek sūtīti dati ar zonas ID un pieprasījuma veidu (ieslēgt/izslēgt).3. Tiek pārbaudīts kopējais drošības sistēmas stāvoklis (ieslēgta/daļēji ieslēgta/izslēgta).
Izvade
<ol style="list-style-type: none">1. Kad tiek saņemti dati no aizmugursistēmas ar īpašuma zonas ID un izmainīto stāvokli, šie dati tiek piesaistīti atbilstošās īpašuma zonas rindai un dati tiek attēloti.2. Atkarībā no kopējā drošības sistēmas stāvokļa, tiek iespējotas/atspējotas drošības sistēmas kopējās ieslēgšanas/izslēgšanas iespējas.3. Ja no aizmugursistēmas tiek saņemts signāls par to, ka zonā, kurā signalizācija ieslēgta, ir bijusi kustība, papildus tam, ka signalizācija sāk skanēt, izvēlnē drošības kontroles taustiņš iezīmējas sarkans.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

Identificējums – FP11

Ievads
Funkcijas mērķis ir ļaut lietotājam ieslēgt/izslēgt visu drošības sistēmu.
Ievades dati
<ul style="list-style-type: none">• Pieprasījums ieslēgt signalizāciju visās īpašuma zonās – pieprasījumu iespējams veikt tikai pie nosacījuma, ka ir vismaz viena īpašuma zona, kurā signalizācija nav ieslēgta;• Pieprasījums izslēgt signalizāciju visās īpašuma zonās – pieprasījumu iespējams veikt tikai pie nosacījuma, ka ir vismaz viena īpašuma zona, kurā signalizācija ir ieslēgta;
Apstrāde
<ol style="list-style-type: none">1. Tiek izsaukts atbilstošais rest/securitysystem/securitysystem/{mode}/{house_id} tīmekļa serviss ar GET metodi.2. Drošības zonām tiek piesaistīts saņemtais JSON fails.
Izvade
<ol style="list-style-type: none">1. Visas drošības zonas attēlojas attiecīgi kā ieslēgtas/izslēgtas.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

2.3.12. Īpašuma gaismas slēdžu kontrole

Identificējums – FP12

Ievads
Šīs funkcijas mērķis ir nodrošināt attālinātu gaismas slēdžu kontroli.
Ievades dati
1. Pieprasījums ieslēgt/izslēgt kādu īpašuma gaismas avotu.
Apstrāde
<ol style="list-style-type: none">1. Izmantojot datu piesaisti, tiek iegūts gaismas avota ID un pieprasījuma veids.2. Šie dati tiek sūtīti serverim, izmantojot rest/lights/light tīmekļa servisu ar PUT metodi.3. Saņemot atbildi no servera, tiek veikta kopējā gaismas avotu stāvokļa pārbaude (neviens/daļa no visiem/visi gaismas avoti ieslēgti).
Izvade
<ul style="list-style-type: none">• Kad tiek saņemti dati no aizmugursistēmas ar īpašuma gaismas avota ID un izmainīto stāvokli, šie dati tiek piesaistīti atbilstošā īpašuma gaismas objekta rindai un jaunie dati tiek attēloti.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

2.3.13. Visu gaismas slēdžu kopējā kontrole

Identificējums – FP13

Ievads
Funkcijas mērķis ir ļaut lietotājam ieslēgt/izslēgt visas īpašumā esošās gaismas reizē.
Ievades dati
<ul style="list-style-type: none">• Pieprasījums ieslēgt visus īpašuma gaismas avotus – pieprasījumu iespējams veikt tikai pie nosacījuma, ka ir vismaz viens gaismas avots, kas ir izslēgts;• Pieprasījums izslēgt visus īpašuma gaismas avotus – pieprasījumu iespējams veikt tikai pie nosacījuma, ka ir vismaz viens gaismas avots, kas ir ieslēgts;
Apstrāde
<ol style="list-style-type: none">1. Tiek izsaukts pieprasījuma veidam (ieslēgt/izslēgt) atbilstošais rest/lights/lights/{mode}/{house_id} tīmekļa serviss ar GET metodi.2. Gaismas objektiem tiek piesaistīts atbildē saņemtais JSON fails.
Izvade
<ol style="list-style-type: none">1. Visi gaismas avoti attēlojas attiecīgi kā ieslēgti/izslēgti.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies.

Identificējums – FP14

Ievads
Funkcijas mērķis ir nodrošināt lietotājam iespēju redzēt izvēlēta sensora mērījumus noteiktā laika periodā grafikā.
Ievades dati
<ul style="list-style-type: none"> • Izvēlēts sensors no izvērstās izvēlnes, kura mērījumus noteiktā laika periodā lietotājs vēlas redzēt. • Laika perioda sākuma un beigu laiki formātā, jāizvēlas gads, mēnesis, datums un laiks, kas satur diennakts stundu un minūti. • Visi lauki ir obligāti aizpildāmi. • Pieprasījums grafika parādīšanai uz ekrāna.
Apstrāde
<ol style="list-style-type: none"> 1. No lietotāja ievades tiek iegūts sensors, laika perioda sākums un beigas. 2. Dati tiek noformatēti un nosūtīti serverim, izmantojot rest/sensor/sensor/{sensor_id} tīmekļa servisu ar PUT metodi. 3. Izmantojot izvēlēto sensoru, tiek uzstādīta grafika Y ass, kā arī izvēlēts grafika tips: <ul style="list-style-type: none"> • Līnijas – grafikiem ar sensoru mērījumiem skaitliskās vērtībās. • Stabiņu – grafikiem ar sensoru mērījumiem binārās vērtībās. 4. Servera atbildes dati tiek piesaistīti grafikam.
Izvade
<ol style="list-style-type: none"> 1. Kad no servera tiek sagaidīta atbilde, kas satur JSON formāta datu fragmentu ar sensora mērījumiem ar laiku kurā tie veikti, tiek parādīts grafiks ar šo informāciju.
Kļūdu paziņojumi
<ul style="list-style-type: none"> • Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies. • Ja izvēlētajam sensoram izvēlētajā laika periodā nav neviena mērījuma, tiek parādīts dialoglodziņš ar paziņojumu, ka par izvēlēto sensoru nav datu.

- Ja kāds no aizpildāmajiem laukiem nav aizpildīts, tiek parādīts attiecīgs paziņojums un šie lauki tiek iezīmēti sarkanā krāsā.

2.3.15. Sensoru izvērstās izvēlnes filtrēšana

Identificējums – FP15

Ievads
Funkcijas mērķis ir ļaut lietotājam atlasīt objektus (sensorus), kas parādās izvērstajā izvēlnē, lai vieglāk atrastu sev interesējošo objektu.
Ievades dati
1. Filtra izvēle.
Apstrāde
<ol style="list-style-type: none"> 1. Izmantojot datu piesaisti, tiek iegūti visi izvērstās izvēlnes objekti un izvēlētā filtra atslēgas atribūts. 2. Atbilstošais sensora tipa un adreses(nosaka, ko sensors mēra, ja tas veic dažādu veidu mērījumus) filtrs tiek pielāgots izvērstās izvēlnes objektiem.
Izvade
1. Izvēloties filtru no izvērstās izvēlnes, jāizvēlas tikai starp tā tipa sensoriem, kurus atlasījis izvēlētais filtrs.
Kļūdu paziņojumi
-

2.3.16. Valodas maiņa un uzstādīšana

Identificējums – FP16

Ievads
Funkcijas mērķis ir mainīt skatos esošos tekstus pret tiem pašiem tekstiem citā valodā, kā arī ļaut lietotājam uzstādīt valodu, kurā turpmāk lietot aplikāciju.
Ievades dati
1. Valodas izvēle.
Apstrāde
1. Tiek nomainīta aplikācijas valoda. 2. Tiek nomainīta lietotājam datu bāzē piesaistītā valoda uz valodu, kas tika izvēlēta.
Izvade
1. Teksti visos skatos tiek nomainīti pret tiem pašiem tekstiem valodā, ko lietotājs izvēlējies. 2. Nākamreiz, kad lietotājs autorizēsies, visi teksti tiks parādīti lietotājam datu bāzē piesaistītajā valodā.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies, un valoda tiek nomainīta tikai tā brīža sesijā.

2.3.17. Ventilācijas CO2 līmeņa uzstādīšana

Identificējums – FP17

Ievads
Funkcijas mērķis ir ļaut lietotājam uzstādīt CO2 līmeni, kas ventilācijai ir jāsasniedz, kā arī attēlot pašreiz uzstādīto CO2 līmeni.
Ievades dati
CO2 līmenis – vesels skaitlis, kas ir robežās no 0 – 1000. Šī validācija notiek gan lietotnes klienta pusē, gan servera pusē.
Apstrāde
<ol style="list-style-type: none">1. Dati tiek nosūtīti serverim, izmantojot rest/ventilationsystem/ventilationsystem/co2 tīmekļa servisu.2. Atbildes dati ar nomainīto uzstādīto CO2 līmeni tiek piesaistīti skatam.
Izvade
<ol style="list-style-type: none">1. Parādās paziņojums, ka CO2 līmenis ir veiksmīgi nomainīts.2. Kā uzstādītais CO2 līmenis tiek uzrādīts nomainītais CO2 līmenis.
Kļūdu paziņojumi
<ul style="list-style-type: none">• Ja REST servisa izsaukums ir neveiksmīgs (padoti nepareizi dati/savienojums ar serveri nav izdevies), tiek parādīts dialoglodziņš ar paziņojumu, ka datu apmaiņa nav izdevusies, un valoda tiek nomainīta tikai tā brīža sesijā.

2.4. Nefunkcionālās prasības

2.4.1. Veiktspējas prasības

Sistēmai jāspēj nodrošināt vairāku lietotāju vienlaicīgu piekļuvi un darbošanos ar sistēmu.

2.4.2. Drošības prasības

- Iebūvēta aizsardzība pret nesankcionētu lietotāju piekļuvi sistēmai.
- Lietotāju paroli glabāšana datu bāzē šifrētā veidā.

2.4.3. Pieejamība

Sistēmai jābūt pieejamai visas dienas nedēļā. Reizi mēnesī var būt iespējama sistēmas uzlabošana un datubāzes dublēšanas darbības, kuru laikā sistēma var būt nepieejama.

2.4.4. Uzturamība

Programmprodukts ir veidots, izmantojot MVC arhitektūras šablonu, tāpēc vajadzības gadījumā sistēma ir viegli papildināma ar jauniem skatiem, laukiem un funkcionalitātēm.

2.5. Ārējās saskarnes prasības

- Lietotāja saskarnei ir jābūt ērti lietojamai gan datorā, gan mobilās ierīcēs.
- Pogām ir jābūt iespējamām tikai tad, kad ar tām iespējams veikt paredzēto funkciju.
- No visiem skatiem ir jābūt iespējai izrakstīties.

- Izmantojot lietotni datora ekrānā, no jebkura skata ir jābūt iespējai pārslēgties uz citu lietotnes funkcionalitāti.
- Lietotājam ir jābūt informētam kļūdu gadījumos.
- No jebkura skata ir jāvar pārslēgties uz citu valodu.

3. POGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

3.1. Ievads

3.1.1. Nolūks

Programmatūras projektējuma apraksta nolūks ir attēlot, kā PPS definētās prasības ir implementētas. Tam ir jāattēlo programmaprodukta projektējums, ko var izmantot produkta plānošanai un implementēšanai.

3.1.2. Darbības sfēra

Lietotnes “Gudrā Māja” mērķis ir nodrošināt attālinātu īpašuma ierīču kontroli un novērot mājas kopējo stāvokli. Projekta “Gudrā Māja” lietotāja saskarnes mērķis ir saprotamā veidā nodrošināt šo kontroļu izmantošanu un informācijas attēlošanu lietotnes klienta daļā.

3.1.3. Saistība ar citiem dokumentiem

Dokuments izstrādāts saskaņā ar standartu „LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.

3.2. Trasējamības tabula

PPS		PPA
Identif.	Nosaukums	Moduļa un metodes nosaukums
FP1 FP3 FP4	Lietotāja reģistrācija; Lietotāja dzēšana; Lietotāja rediģēšana;	Moduļa "Administrator" metodes: <ul style="list-style-type: none"> • "openCreateDialog" • "validateInput" • "createUser" • "onDeletePress" • "openEditDialog" • "editUser" • "setAdminData"
FP2	Lietotāja autentifikācija	Moduļa "Session" metodes: <ul style="list-style-type: none"> • "login"
FP5 FP6	Apkures sistēmas trīszaru vārstu temperatūras uzstādīšana; Apkures sistēmas izslēgšana;	Moduļa "Heating" metodes: <ul style="list-style-type: none"> • "changeTemperature" • "turnSystemOff" • "onOrOff" • "setHeatingData"
FP7 FP17	Ventilācijas regulēšana	Moduļa "Ventilation" metodes: <ul style="list-style-type: none"> • "setVentilatorPower" • "setCO2Level" • "setVentilationData"
FP8 FP9	Vārtu kontrole, vārtu apturēšana	Moduļa "Gates" metodes: <ul style="list-style-type: none"> • "gateStatePress" • "gateStopPress" • "setGatesData"
FP10 FP11	Drošības sistēmas kontrole	Moduļa "Security" metodes: <ul style="list-style-type: none"> • "changeZoneMode"

		<ul style="list-style-type: none"> • “changeSystemMode” • “systemModeFunction” • “setSecurityData”
FP12 FP13	<p>Īpašuma gaismas slēdžu kontrole;</p> <p>Visu gaismas slēdžu kopējā kontrole;</p>	<p>Moduļa “Light” metodes:</p> <ul style="list-style-type: none"> • “changeLightState” • “changeSystemState” • “systemStateFunction” • “setLightData”
FP14 FP15	<p>Sensoru mērījumu pārskata attēlošana;</p> <p>Sensoru izvērstās izvēlnes filtrēšana;</p>	<p>Moduļa “Overview” metodes:</p> <ul style="list-style-type: none"> • “setAxisValues” • “makeChart” • “showChartOverview” • “handleIconTabBarSelect” • “generateFilter”
FP16	Valodas maiņa un uzstādīšana	<p>Moduļa “Language” metodes:</p> <ul style="list-style-type: none"> • “setUserLanguage”

3.3. Moduļu dekompozīcija

3.3.1. Administrator modulis

Šis modulis atbild par darbību ar lietotājiem veikšanu, kas ir funkcionalitāte, kas pieejama tikai lietotājiem

Metodes nosaukums	Apraksts
“openEditDialog”	Atver uznirstošo logu, kurā attēlota lietotāja informācija, kuru administrators var mainīt. Pēc lietotāja rediģēšanas aizver uznirstošo logu un atjauno lietotāju tabulas datus, lai izmainītie dati ir redzami.
“validateInput”	Nodrošina kļūdu paziņojumu parādīšanu un ievades lauku iezīmēšanu parādīšanu nepareizu datu ievades gadījumā.
“createUser”	Pēc ievades validācijas izveido JSON faila fragmentu, kas tiek nosūtīts serverim, kur tiek izveidots jauns lietotājs.
“onDeletePress”	Atver dialoglogu, kas pieprasa apstiprinājumu par lietotāja dzēšanu. Pēc lietotāja dzēšanas apstiprinājuma nosūta serverim pieprasījumu dzēst lietotāju. Pēc veiksmīgas lietotāja dzēšanas atjauno lietotāju tabulas datus, lai izdzēstais lietotājs tabulā netiktu attēlots.
“openCreateDialog”	Atver uznirstošo logu, kurā iespējams izveidot jaunu lietotāju. Pēc lietotāja rediģēšanas aizver uznirstošo logu un atjauno lietotāju tabulas datus, lai izveidotais lietotājs ir redzams.
“editUser”	Pēc ievades validācijas izveido JSON faila fragmentu, kas tiek nosūtīts serverim, kur lietotāja dati tiek izmainīti.
“setAdminData”	Piesaista skatam no servera iegūtos, tikai administratoram pieejamos lietotājus aprakstošos datus..

3.3.2. *Session* modulis

Šis modulis atbild par lietotāja autentificēšanu.

Metodes nosaukums	Apraksts
“logIn”	Nosūta serverim lietotāja ievadīto informāciju pareizu datu ievades gadījumā, kur lietotājs tiek autentificēts. Nepareizu datu ievades gadījumā parāda kļūdas paziņojumu.

3.3.3. *Heating* modulis

Šis modulis atbild par trīszaru vārstu temperatūras regulēšanu un apkures sistēmas izslēgšanu.

Metodes nosaukums	Apraksts
“changeTemperature”	Validē lietotāja ievadi, pareizas ievades gadījumā izveido JSON failu, nosūta to serverim, kur tiek izmainīta trīszaru vārsta uzstādītā temperatūra. Atjauno konkrētā trīszaru vārsta datus, lai uzrādītos nomainītā temperatūra.
“turnSystemOff”	Nosūta serverim pieprasījumu uzstādīt visu trīszaru vārstu temperatūras uz 0°C.
“onOrOff”	Iespējo vai atspējo apkures sistēmas izslēgšanu atkarībā no trīszaru vārstu uzstādītās temperatūras.
“setHeatingData”	Iegūst no servera apkures sistēmas aprakstošos datus un attēlo tos.

3.3.4. *Ventilation* modulis

Šis modulis atbild par ventilatoru režīmu regulēšanu.

Metodes nosaukums	Apraksts
“setVentilatorPower”	No lietotāja ievades izveido JSON formāta failu, ko sūta serverim, kur tiek atbilstoši nomainīts ventilatora režīms. Attēlo nomainīto ventilatora režīmu.
“setCO2Level”	Validē lietotāja ievadi, izveido JSON formāta fragmentu, ko sūta serverim, kur tiek nomainīts uzstādītais CO2 līmenis. Attēlo nomainīto CO2 līmeni.
“setVentilationData”	Iegūst no servera ventilācijas sistēmu aprakstošos datus un attēlo tos.

3.3.5. *Gates* modulis

Metodes nosaukums	Apraksts
“gateStatePress”	Pēc lietotāja pieprasījuma izveido JSON failu, ko sūta serverim, kur vārti tiek atvērti/aizvērti. Attēlo vārtu starpstāvokli, kamēr tie aizveras/atveras un parāda gala stāvokli, kad darbība pabeigta.
“gateStopPress”	Pēc lietotāja pieprasījuma izveido JSON failu, ko sūta serverim, kur vārti tiek apturēti. Parāda vārtu stāvokli.
“setGatesData”	Iegūst no servera īpašumā esošo vārtu aprakstošos datus un attēlo tos.

3.3.6. *Security* modulis

Metodes nosaukums	Apraksts
“changeZoneMode”	No lietotāja pieprasījuma izveido JSON failu, ko sūta serverim, kur tiek izmainīts drošības zonas stāvoklis. Attēlo izmainīto drošības zonas stāvokli.
“changeSystemMode”	Sūta serverim lietotāja pieprasījumu ieslēgt/izslēgt drošības sistēmu. Attēlo drošības zonu un drošības sistēmas izmainītos stāvokļus.
“systemModeFunction”	Attēlo drošības sistēmas stāvokli.
“setSecurityData”	Iegūst no servera drošības sistēmu aprakstošos datus un attēlo tos.

3.3.7. *Light* modulis

Metodes nosaukums	Apraksts
“changeLightState”	Pēc lietotāja pieprasījuma izveido JSON failu, ko sūta serverim, kur tiek nomainīts gaismas avota stāvoklis. Attēlo gaismas avota izmainīto stāvokli.
“changeSystemState”	Sūta serverim lietotāja pieprasījumu ieslēgt/izslēgt visus gaismas avotus. Attēlo gaismas objektu izmainītos datus.
“systemStateFunction”	Attēlo gaismas avotu kopējo stāvokli.
“setLightData”	Iegūst no servera īpašumā esošo gaismas avotu aprakstošo informāciju un attēlo to.

3.3.8. *Overview* modulis

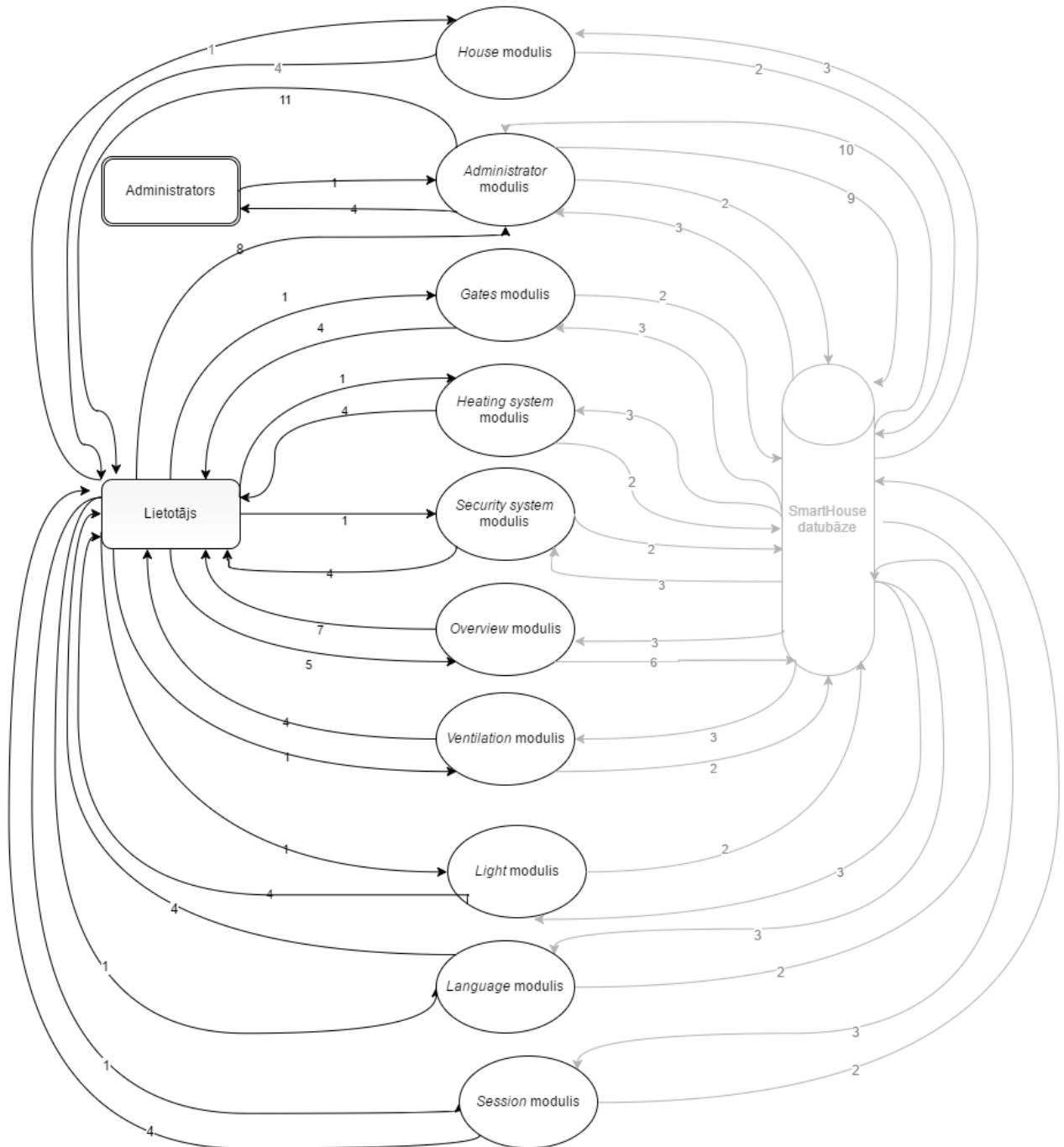
Metodes nosaukums	Apraksts
“setAxisNames”	Ielādējot sensoru mērījumu pārskata lapu, attēlo tukšu grafiku ar nenosauktām X un Y asīm.
“makeChart”	Definē grafikam visus iespējamus X un Y asu nosaukumus.
“setAxisValues”	Atkarībā no tā, kurš sensors izvēlēts, uzstāda: Grafika Y asi; Grafika X asi; Grafika tipu;
“showChartOverview”	No lietotāja ievades izveido JSON formāta failu, kas pieprasa nepieciešamo informāciju. Izmantojot metodes “makeChart” un “setAxisValues”, izveido grafiku, kam piesaista datus, kas saņemti no servera.
“handleIconTabBarSelect”	Uzstāda īpašuma sensoru izvērstajai izvēlnei lietotāja izvēlēto filtru, lai tiktu attēloti tikai izvēlēta tipa sensori.
“generateFilter”	Izveido filtru sensoriem, kas veic divu tipu mērījumus(piem., mēra gan temperatūru, gan mitrumu).

3.3.9. *Language* modulis

Metodes nosaukums	Apraksts
“setUserLanguage”	Pēc lietotāja pieprasījuma nomaina pašreizējās sesijas valodu, kā arī izveido JSON failu, ko sūta serverim, kur tiek nomainīta lietotājam piesaistītā valoda.

3.3. Atkarības apraksts

3.3.1. Pirmā līmeņa datu plūsmu diagramma

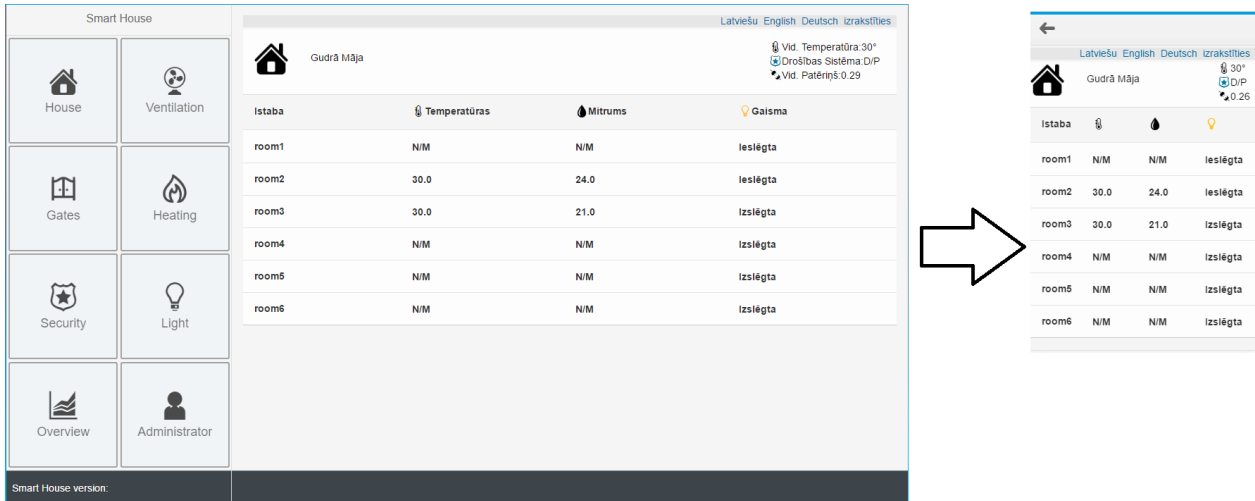


Attēls 2

1. Nomaināmie dati; Datu apskates pieprasījums;
2. Izmaiņu saglabāšana; Datu pieprasījums;
3. Pieprasītie dati;
4. Aprakstošo datu attēlojums; Izmainīto datu attēlojums;
5. Konkrēta pārskata pieprasījums;
6. Datu pieprasījums;
7. Konkrētā pārskata attēlojums.
8. Autorizācijas dati; Autorizācijas pieprasījums;
9. Autorizācijas datu pārbaude;
10. Autorizācijas apstiprinājums;
11. Autorizācija;

3.3. Ārējā saskarne

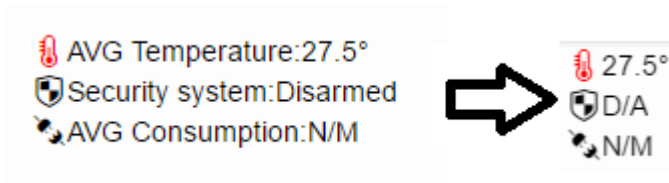
3.3.1. Ārējās saskarnes pielāgošanās ierīcei un ekrāna izmēram



Attēls 3

Lietotāja saskarne, pielāgojoties ierīcei vai mazākam ekrānam:

- Slēpj funkciju izvēles skatu mazākos ekrānos – šādā gadījumā atpakaļ uz izvēles skatu var nokļūt ar navigācijas pogu augšējā kreisajā stūrī (piem. Attēls 3);
- Maina elementu izmērus (platumu, garumu) (piem. Attēls 3);
- Nostāda elementus vienu zem otra, ja nepieciešams (piem. Attēls 6);
- Sabīda kolonnas vienu zem otras/noslēpj mazāk svarīgas kolonnas (piem. Attēls 5);
- Slēpj vai saīsina tekstu, kas neietilpst ekrānā (piem. Attēls 4);



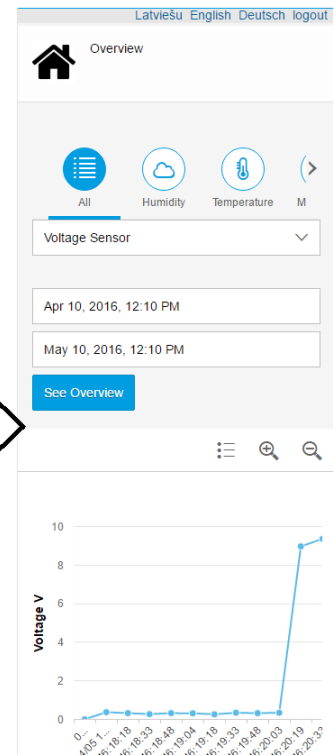
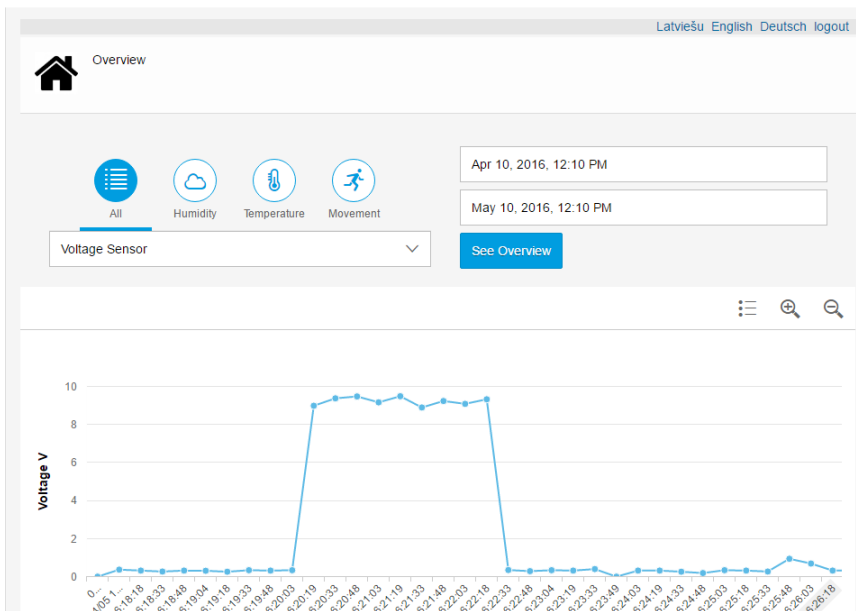
Attēls 4

Gates	State	Switch	Emergency
GarageGate	Open	<input type="button" value="Open"/> <input type="button" value="Close"/>	<input type="button" value="Stop!"/>



Gates	Switch
GarageGate	<input type="button" value="Open"/> <input type="button" value="Close"/>
Emergency:	<input type="button" value="Stop!"/>

Attēls 5



Attēls 6

3.3.1. Valodas izvēle

Latviešu English Deutsch logout

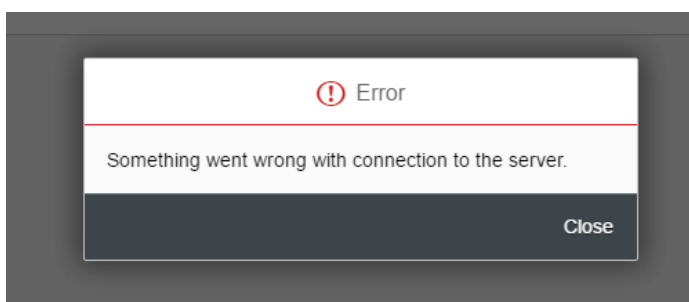
Attēls 7

Katra skata augšpusē ir josla, kurā iespējams mainīt to, kādā valodā tiek rādīts lietotnes skatu teksts. Kad lietotājs nospiež kādu no šīm valodām, teksti pārmainās uz attiecīgo valodu.

Šajā joslā arī iespējams izrakstīties, lai tas būtu iespējams no katra skata.

3.3.2. Lietotnes vispārējie kļūdu paziņojumi

Ja REST servisa izsaukumā ir kļūda un netiek saņemta atbilde/dati, tiek parādīts paziņojums par to, ka savienojums ar serveri nav izdevies (skat. Attēls 8).



Attēls 8

3.3.2. Autorizācijas skats

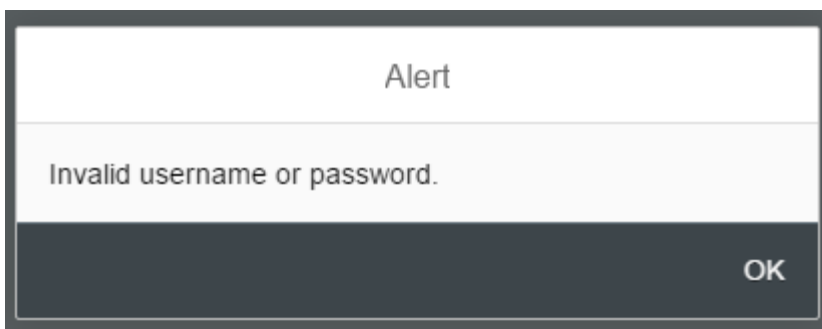
Atsauce uz funkcionālo prasību: FP2



Attēls 9

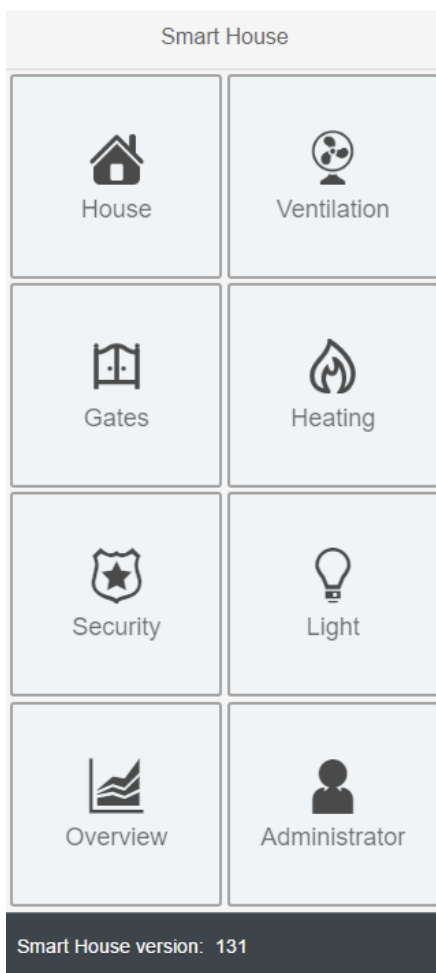
Autorizācijas skats, kurā, ievadot pareizu lietotājvārdu un paroli, iespējams autorizēties, lai izmantotu lietotnes funkcijas.

Ja tiek veikts pieprasījums autorizēties ar lietotājvārda un paroles kombināciju, kas datu bāzē neeksistē, tiek parādīts paziņojums, ka lietotājvārds vai parole ir nepareizi ievadīts (skat. Attēls 10)



Attēls 10

3.3.3. Funkciju izvēles skats

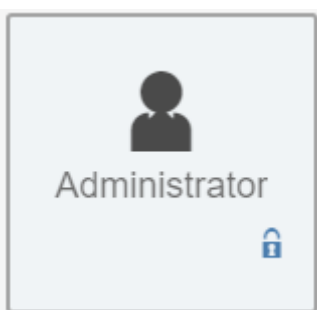


Attēls 11

Lietotnes funkciju skats, kurā, noklikšķinot uz kāda no laukiem, iespējams nokļūt atbilstošajā skatā.

Ja lietotni izmanto datorā, šis skats vienmēr atrodas cita skata kreisajā pusē, lai no jebkura skata var ērti pārvietoties uz citu.

Ja lietotni izmanto telefonā, kā sākuma ekrāns tiek parādīts šis skats.



Ja lietotni izmanto lietotājs, kas nav administrators, iespēja nokļūt skatā, kur var veikt administratoram paredzēto funkcionalitāti, tiek uzrādīta kā aizslēgta un nereaģē uz pieprasījumu nokļūt skatā. (skat. Attēls 12)

Attēls 12

3.3.4. Mājas kopējā stāvokļa skats

A screenshot of a web application interface for a smart home system. On the left is a vertical sidebar with eight buttons: House, Ventilation, Gates, Heating, Security, Light, Overview, and Administrator. The main area shows a header with a home icon, the text "Smart House", and language options (Latviešu, English, Deutsch) and a logout link. Below the header is a table with columns for Room, Temperature, Humidity, and Light. The table contains six rows of data for rooms 1 through 6. At the bottom left of the main area, it says "Smart House version:". At the top right of the main area, there are status indicators: "AVG Temperature:25°", "Security system:D/A", and "AVG Consumption:0.19".

Room	Temperature	Humidity	Light
room1	N/M	N/M	Off
room2	25.0	24.0	Off
room3	25.0	22.0	On
room4	N/M	N/M	Off
room5	N/M	N/M	Off
room6	N/M	N/M	Off

Attēls 13

Ja lietotni izmanto datorā, kā sākuma ekrāns tiek parādīts šis skats.

Mājas kopējā stāvokļa skats, kurā tiek parādīta informācija par istabām:

- Katras istabas vidējā ar sensoriem mērītā vidējā temperatūra;
- Katras istabas vidējā ar sensoriem mērītais vidējais mitrums;
- Katrai istabai tiek parādīts, vai vismaz viena gaisma tajā ir ieslēgta (On – vismaz viena ieslēgta/Off – visas izslēgtas);

Tiek parādīta informācija par māju:

- Visas mājas sensoru mērījumu vidējā temperatūra;
- Drošības sistēmas stāvoklis (Ieslēgta/Daļēji ieslēgta/Izslēgta);
- Nomērītais vidējais mājas strāvas patēriņš;

Lai padarītu mājas kopējo stāvokli uzskatāmāku:

- Termometra ikona tiek attēlota sarkanā krāsā, ja apkures sistēma ieslēgta (vismaz viens trīszaru vārsts uzstādīts uz temperatūru virs 0°C), pretējā gadījumā – melnā krāsā.
- Drošības sistēmas ikona tiek attēlota gaiši zilā krāsā, ja drošības sistēma ir ieslēgta (visām drošības zonām ieslēgta signalizācija), tumši zilā krāsā, ja drošības sistēma daļēji ieslēgta (vismaz vienai drošības zonai izslēgta signalizācija un vismaz vienai - ieslēgta), melnā, ja drošības sistēma izslēgta.
- Spuldzītes ikona tiek attēlota dzeltenā krāsā, ja vismaz viena gaisma mājā ir ieslēgta, pretējā gadījumā – melnā.

3.3.5. Ventilācijas skats

Sensor	Type	Humidity
Gass	G	218.0
DH22	C	28.0

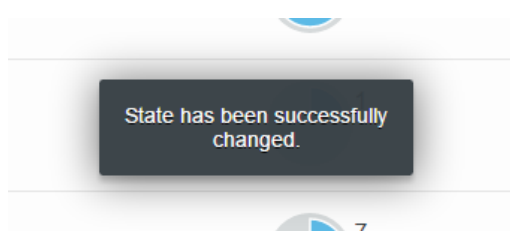
Attēls 14

Atsauce uz funkcionālo prasību: FP7

Ventilācijas kontrolei paredzētais skats, kurā iespējams regulēt ventilatoru režīmus.(skat. Attēls 14)

Ar slēdži regulējot ventilatora režīmu, mainās arī vērtība attiecīgā ventilatora diagrammā, kas attēlo režīmu. Režīms tiek nomainīts, kad slēdzis tiek atlaists.

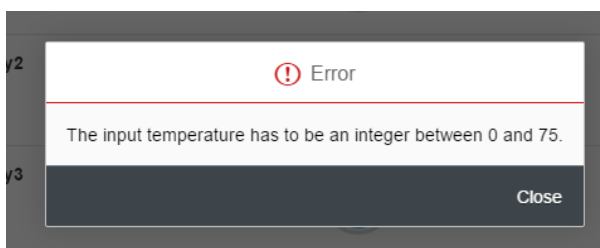
Kad ventilatora režīms tiek nomainīts tiek uzrādīts attiecīgais paziņojums (skat. Attēls 15)



Attēls 15

Iespējams uzstādīt CO2 līmeni, kāds ventilācijas sistēmai ir jāsasniedz.

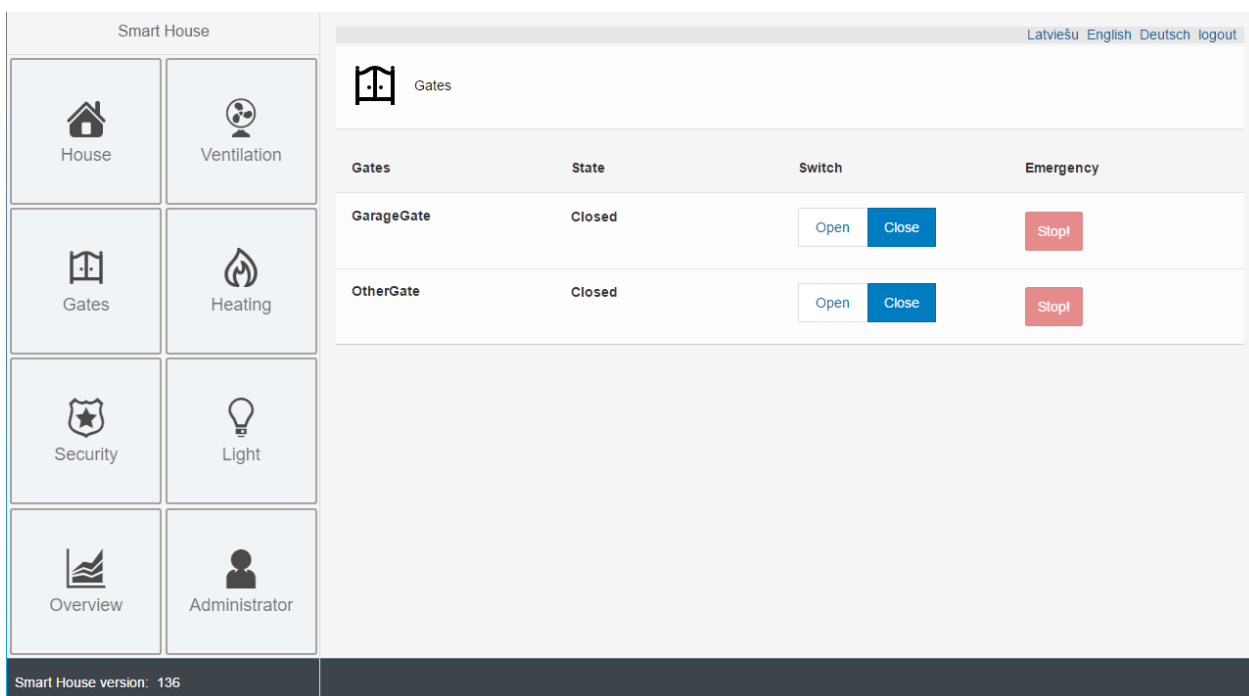
Ja tiek ievadīta vērtība, kas nav vesels skaitlis vērtībā no 1 līdz 1000, tiek parādīts uznirstošais logs ar attiecīgo paziņojumu. (Attēls 16)



Attēls 16

Tiek parādīti ventilācijas sistēmas sensoru mērījumi, lai lietotājs varētu gūt priekšstatu par pašreizējo ventilācijas sistēmas stāvokli.

3.3.6. Vārtu kontroles skats



Attēls 17

Atsauce uz funkcionālo prasību: FP8, FP9;

Vārtu kontrolei paredzētais skats, kurā iespējams atvērt un aizvērt īpašumā esošos vārtus.

Tiek uzrādīti visi īpašumā esošie vārti ar to nosaukumu un pašreizējo stāvokli.

Ja vārti ir atvērti/aizvērti, tad poga, kas paredzēta attiecīgi atvēršanai/aizvēršanai, ir atspējota, kā arī poga, kas paredzēta vārtu apturēšanai ārkārtas situācijās, ir atspējota.

Ja vārti ir apturēti starpstāvoklī (atvērti/aizvērti), tad gan atvēršanas, gan aizvēršanas pogas ir iespējotas, bet apturēšanas poga ir atspējota.

Kad tiek nospiesta atvēršanas/aizvēršanas poga, sākas attiecīgais process, attiecīgi tiek nomainīts pašreizējais vārtu stāvoklis (atveras/aizveras), un gan atvēršanas, gan aizvēršanas poga attiecīgajiem vārtiem tiek atspējota, savukārt apturēšanas poga tiek iespējota, lai neparedzētā situācijā varētu vārtus apturēt.

3.3.7. Apkures sistēmas kontroles skats

	Current temperature	Now set to	Set to
trigeminalvalve1	0.0	66	<input type="text"/>
trigeminalvalve2	0.0	66	<input type="text"/>
trigeminalvalve3	0.0	66	<input type="text"/>
trigeminalvalve4	0.0	66	<input type="text"/>

Sensor	Temperature	Description
temperature1	0.0	temperature1

Attēls 18

Atsauce uz funkcionālo prasību: FP5, FP6;

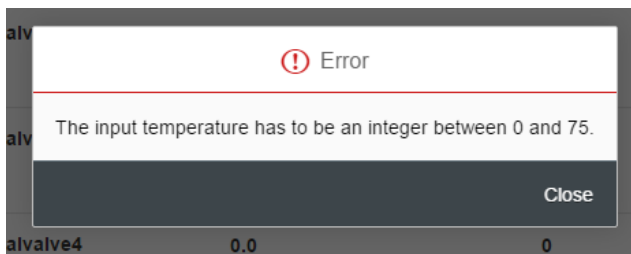
Apkures sistēmas kontrolei paredzētais skats.

Tiek uzrādīti īpašumā esošie trīszaru vārsti, to pašreizējā temperatūra un tas, cik grādu celsijā ir jāsasniedz katram trīszaru vārstam.

Tiek uzrādīti apkures sistēmas sensoru mērījumi, lai radītu priekšstatu par apkures sistēmas stāvokli.

Lietotājs var uzstādīt trīszaru vārstu temperatūru pēc savām vajadzībām robežās no 0 līdz 75°C.

Ja lietotājs mēģina uzstādīt temperatūru, kas nav vesels skaitlis robežās no 0 līdz 75, tiek parādīts uznirstošs logs ar attiecīgo informāciju. (skat. Attēls 19)

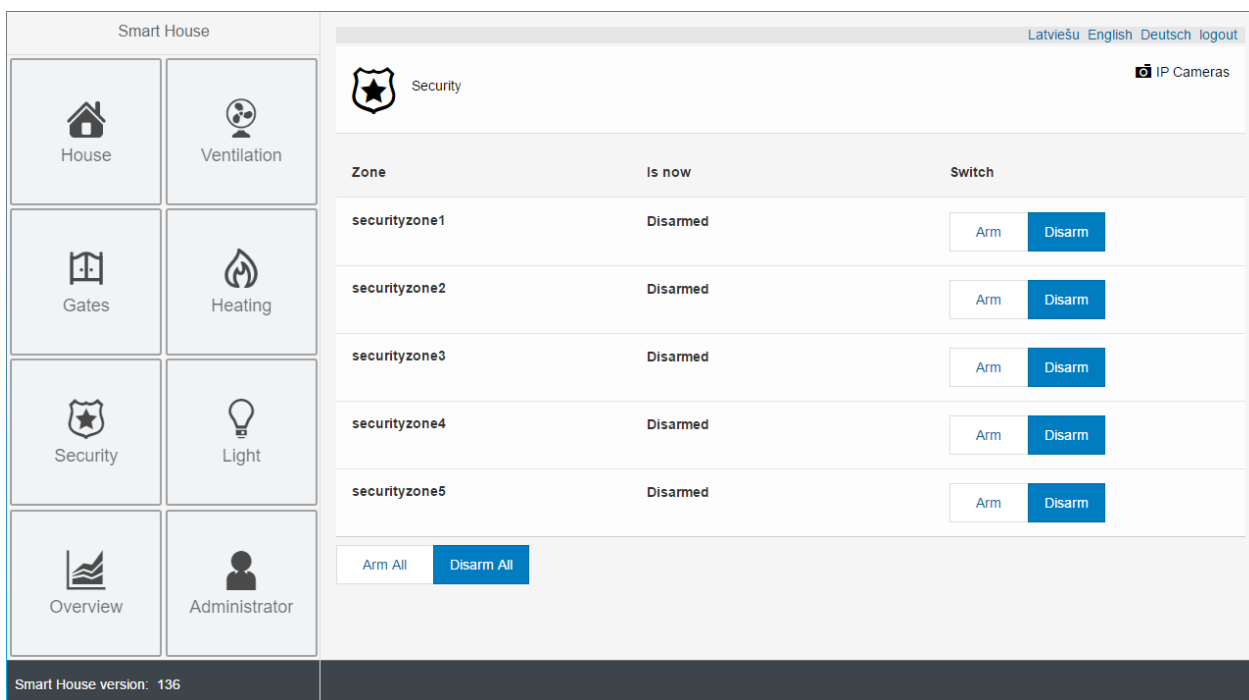


Attēls 19

Lietotājs var izslēgt apkures sistēmu (uzstādīt visu trīszaru vārstu temperatūru uz 0°C).

Izslēgšanas poga ir iespējota tikai tad, ja vismaz viena trīszaru vārsta temperatūra ir uzstādīta virs 0°C.

3.3.8. Drošības sistēmas kontroles skats



Attēls 20

Atsauce uz funkcionālo prasību: FP10, FP11;

Drošības sistēmas kontrolei paredzētais skats.

Tiek uzrādītas visas drošības sistēmas ar attiecīgo pašreizējo stāvokli (signalizācija ieslēgta/izslēgta).

Tiek uzrādīts drošības sistēmas kopējais stāvoklis.

Drošības zonās iespējams ieslēgt un izslēgt signalizāciju ar attiecīgajām pogām.

Drošības zonu signalizācijas ieslēgšanas vai izslēgšanas poga ir atspējota atkarībā no tā, vai drošības zonā signalizācija jau ieslēgta vai izslēgta.

Iespējams ieslēgt/izslēgt visu drošības sistēmu, kas nozīmē, ka visām drošības zonām attiecīgi ir ieslēgta vai izslēgta signalizācija.

No šī skata iespējams veikt navigāciju uz skatu, kas satur informāciju par īpašumam piesaistītajām IP kamerām.

Kopējās drošības sistēmas ieslēgšanas un izslēgšanas pogas:

- Atspējota izslēgšanas poga, iespējota ieslēgšanas poga – ja visām drošības zonām atslēgta signalizācija.
- Atspējota ieslēgšanas poga, iespējota izslēgšanas poga – ja visām drošības zonām ieslēgta signalizācija.
- Iespējota gan ieslēgšanas, gan izslēgšanas poga – ja vismaz vienai drošības zonai ir ieslēgta signalizācija un vismaz vienai drošības zonai ir izslēgta signalizācija.

3.3.9. Gaismas kontroles skats

Room	Light	Is now	Switch
kitchen	dummy1	Off	<input type="button" value="On"/> <input type="button" value="Off"/>
bedroom	relay1	Off	<input type="button" value="On"/> <input type="button" value="Off"/>
stairs	relay2	Off	<input type="button" value="On"/> <input type="button" value="Off"/>
kidsroom	dummy2	Off	<input type="button" value="On"/> <input type="button" value="Off"/>
room	dummy4	Off	<input type="button" value="On"/> <input type="button" value="Off"/>
room	dummy5	Off	<input type="button" value="On"/> <input type="button" value="Off"/>

Attēls 21

Atsauce uz funkcionālo prasību: FP12, FP13;

Īpašuma gaismas objektu kontrolei paredzētais skats.

Tiek uzrādīti visi gaismas objekti ar attiecīgo pašreizējo stāvokli(ieslēgts/izslēgts).

Tiek uzrādīts īpašuma kopējais gaismas objektu stāvoklis.

Gaismas objektus iespējams ieslēgt un izslēgt ar attiecīgajām pogām.

Gaismas objekta ieslēgšanas vai izslēgšanas poga ir atspējota atkarībā no tā, vai gaismas objekts jau ieslēgts vai izslēgts.

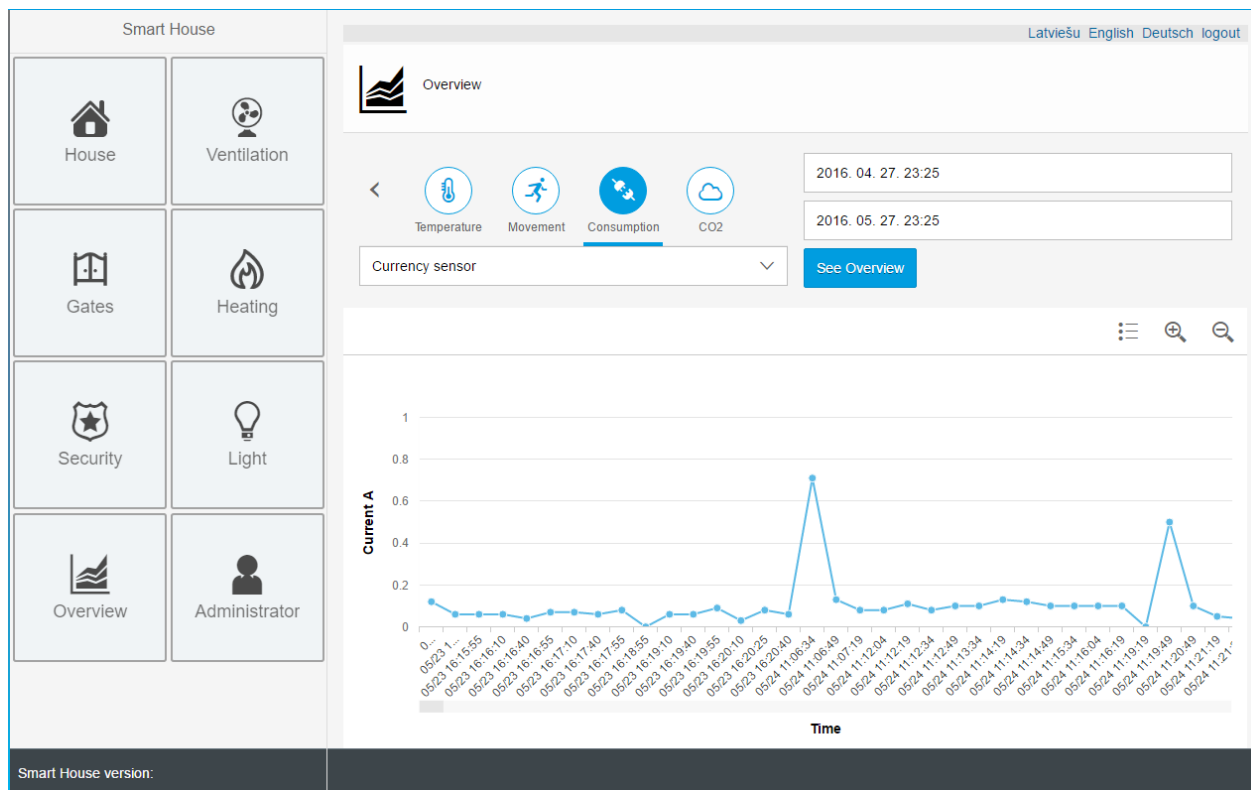
Iespējams ieslēgt/izslēgt visus īpašuma gaismas objektus, kas nozīmē, ka visi gaismas objekti ir attiecīgi ir ieslēgti vai izslēgti.

Kopējās visa īpašuma gaismas objektu ieslēgšanas un izslēgšanas pogas:

- Atspējota izslēgšanas poga, iespējota ieslēgšanas poga – ja visi gaismas objekti ir izslēgti.
- Atspējota ieslēgšanas poga, iespējota izslēgšanas poga – ja visi gaismas objekti ir ieslēgti.

- Iespējota gan ieslēgšanas, gan izslēgšanas poga – ja vismaz viens gaismas objekts ir ieslēgts un vismaz viens gaismas objekts ir izslēgts.

3.3.10. Pārskata skats



Attēls 22

Atsauce uz funkcionālo prasību: FP14, FP15;

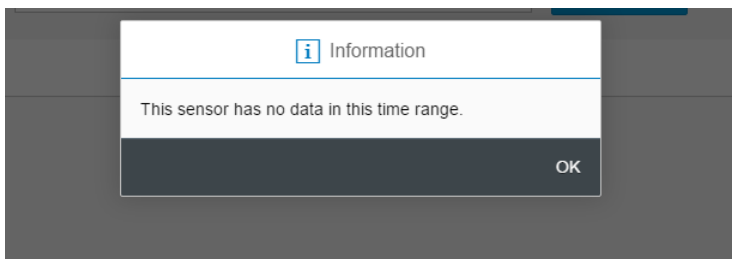
Īpašuma sensoru mērījumu apskatei paredzētais skats.

Pirms grafika parādīšanas:

- Jāizvēlas sensors, par kura mērījumiem lietotājs vēlas redzēt pārskatu. Ja sensoru īpašumā ir daudz, nepieciešamības gadījumā izvērsto izvēlni iespējams filtrēt pēc sensora tipa.
- Jāizvēlas datums, no kura sākot tiks atlasīti izvēlētā sensora mērījumi.
- Jāizvēlas datums, ar kuru beidzot tiks atlasīti izvēlētā sensora mērījumi.

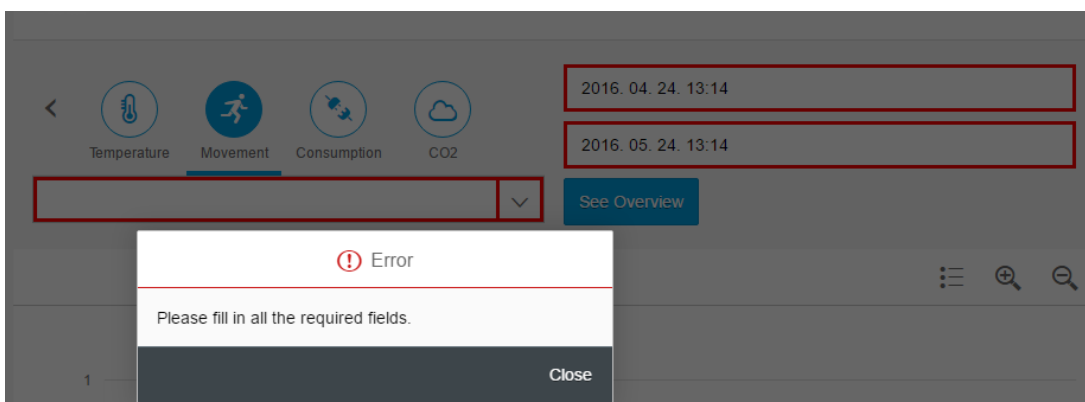
Nospiežot pogu, kas paredzēta grafika parādīšanai, logā, kas atrodas zem sensora un datumu ievades, parādās grafiks ar sensora mērījumiem noteiktajā laika intervālā, ja šajā laika intervālā ir vismaz viens mērījums.

Ja izvēlētajam sensoram norādītajā laika periodā nav neviena mērījuma, par to tiek parādīts paziņojums un grafiks netiek ģenerēts. (skat. Attēls 23)



Attēls 23

Ja kāds no laukiem nav aizpildīts, parādās paziņojums, ka laukiem jābūt aizpildītiem, un visi aizpildāmie lauki tiek iezīmēti sarkanā krāsā. (skat. Attēls 24)



Attēls 24

Grafika Y ass ir sensora mērījuma vērtība, X ass ir laiks formātā MM/DD HH:MM:SS.

Grafiku iespējams pietuvināt un attālināt ar lupas ikonu palīdzību.

3.3.11. Administratora skats

The screenshot shows the 'Smart House' administrator interface. On the left is a navigation menu with icons for House, Ventilation, Gates, Heating, Security, Light, Overview, and Administrator. The main area displays the user management section, including a user profile for 'Administrator' and a table of users.

User	Role	Edit	Delete
Girts Kļava	User	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Gustavs Felsbergs	User	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Rainers Liepiņš	User	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Jānis Bērziņš	User	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
Andris Bērziņš	Administrator	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Buttons:

Smart House version: [unreadable]

Attēls 25

Atsauce uz funkcionālo prasību: FP1, FP3, FP4;

Skats, kas paredzēts administratoram, lai veiktu darbības ar lietotājiem.

Tiek uzrādīti atbilstošās mājas lietotāji un lietotāju lomas.

The screenshot shows the 'Edit A User' form. It contains the following fields:

- Name: Jānis
- Surname: Bērziņš
- Password: Enter a new password
- Confirm: Confirm password
- E-Mail: jb@gmail.com

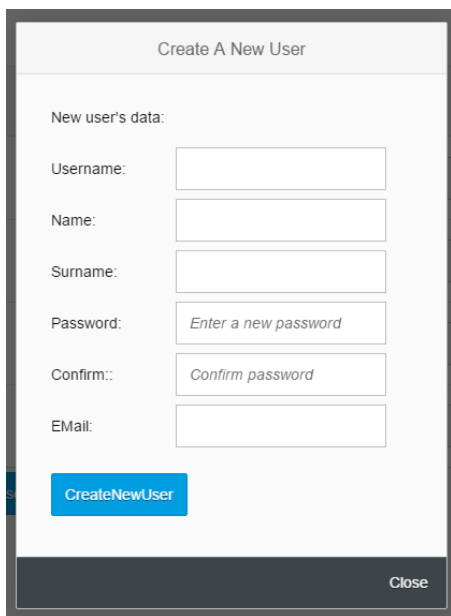
Buttons:

Attēls 26

Nospiežot pogu, kas paredzēta lietotāja rediģēšanai, atveras logs (skat. Attēls 26), kurā var nomainīt:

- Lietotāja vārdu;
- Lietotāja uzvārdu;
- Lietotāja e-pastu;
- Lietotāja paroli;

Nospiežot pogu, kas paredzēta lietotāja dzēšanai, tiek parādīts dialoglodziņš, ka dzēšana notikusi veiksmīgi un tiek uzrādīti lietotāji bez jau izdzēstā lietotāja.



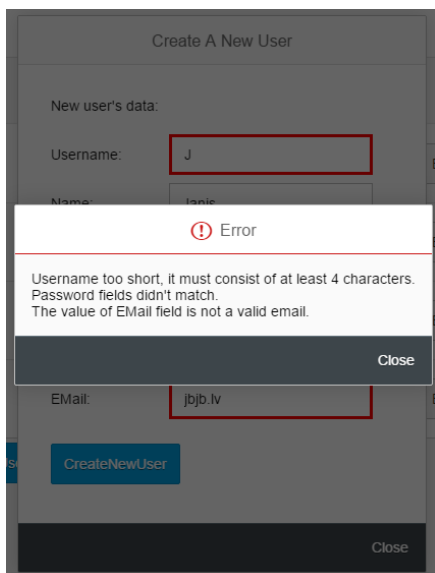
Attēls 27

Nospiežot pogu, kas paredzēta jauna lietotāja izveidei, atveras logs (skat. Attēls 27), kurā jāievada informācija par jauno lietotāju:

- Lietotājvārds;
- Lietotāja vārds;
- Lietotāja uzvārds;
- Lietotāja parole;
- Lietotāja e-pasts;

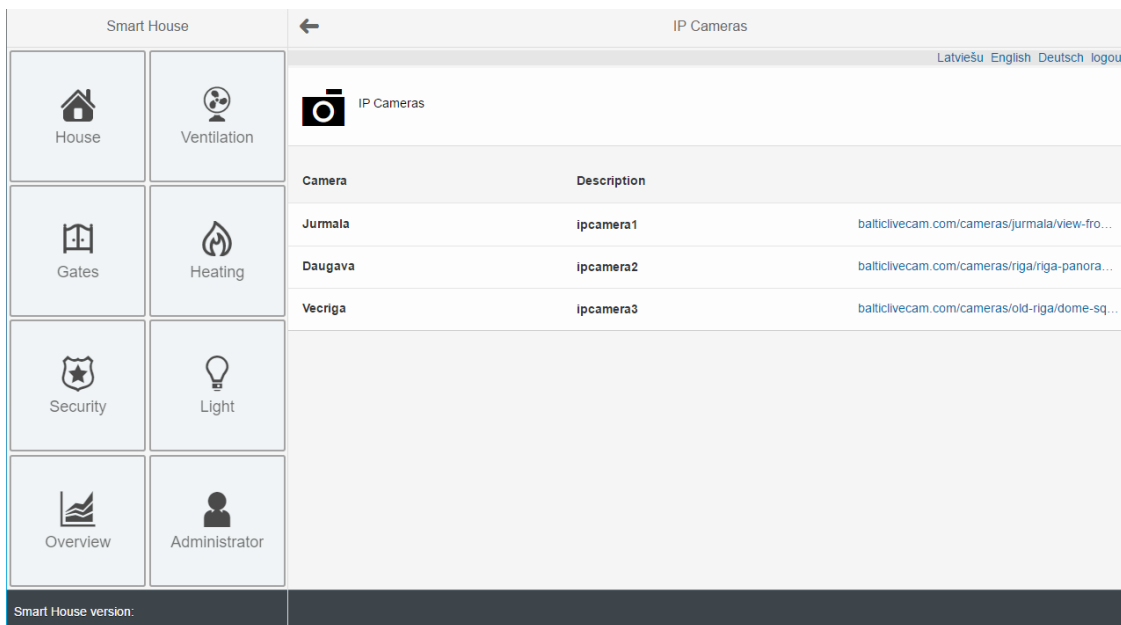
Pēc tam, nospiežot jaunā loga lietotāja izveides apstiprinājuma pogu, tiek izveidots jauns lietotājs, un tiek uzrādīti lietotāji ar izveidoto lietotāju.

Ja lietotāja reģistrācijas vai lietotāja rediģēšanas dialoglogos tiek nepareizi ievadīti dati, tiek parādīts uznirstošs logs ar paziņojumiem par nepareizi ievadītajiem laukiem un nepareizi ievadītie lauki tiek iekrāsoti. (piem. Attēls 28)



Attēls 28

3.3.12. IP Kameru skats



Attēls 29

Skats, kurā tiek uzrādītas visas īpašumam piesaistītās IP kameras, kā arī katras kameras URL, lai varētu ērti redzēt kameras novērojumus.

Ar bultiņas navigācijas pogu nokļūst atpakaļ drošības zonu kontroles skatā.

4. TESTĒŠANAS DOKUMENTĀCIJA

4.1. Ievads

Šis testēšanas plāns apraksta lietotnes „Gudrā Māja” lietotāja saskarnes funkcionalitātes testēšanu. Tā kā šis programmprodukts atbild par to, lai lietotājs būtu informēts par situāciju īpašumā un spētu šo situāciju vajadzības gadījumā mainīt, datiem ir jāuzrādās precīzi un izmaiņu veikšanas gadījumā lietotājam ir jāzina, vai izmaiņas veiksmīgi notikušas.

4.2. Testēšanas žurnāls

4.2.1. Lietotāja reģistrācija

Atsauce uz funkcionālo prasību – FP1

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek pieprasīta lietotāja reģistrēšanas dialogloga atvēršana.	Atveras lietotāja reģistrēšanas dialoglogs.	Izpildās.
Ievadītie dati, kas nepieciešami jauna lietotāja izveidei tiek kļūdaini ievadīti.	Tiek iezīmēti kļūdainie lauki un uz ekrāna tiek parādīts uznirstošais logs, kurā parādās informācija par kļūdainajiem laukiem.	Izpildās.
Tiek ievadīts lietotājvārds, kas jau datu bāzē eksistē.	Tiek parādīts kļūdas paziņojums, ka lietotājs ar šādu lietotājvārdu jau eksistē.	Izpildās.
Visi lauki aizpildīti atbilstoši prasībām un tiek veikts	Uznirstošais logs tiek aizvērts, parādās paziņojums,	Izpildās.

pieprasījums lietotāja izveidei.	ka lietotājs ir veiksmīgi izveidots, un tabulā, kurā uzrādās lietotāji, tiek uzrādīts izveidotais lietotājs.	
Tiek aizpildīts kāds no lietotāja reģistrācijai nepieciešamajiem laukiem.	Ievadot lauku, lauks iezīmējas sarkans, kamēr ievadītā vērtības simbolu skaits ir mazāks par minimālo.	Izpildās.

4.2.2. Lietotāja autentifikācija

Atsauce uz funkcionālo prasību – FP2

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Lietotājvārds un/vai parole nepareizi ievadīts/nav ievadīts.	Parādās skats ar paziņojumu, ka lietotājvārds un/vai parole nav pareizi ievadīts, no kura iespējams nokļūt atpakaļ lietotāja autentifikācijas skatā.	Izpildās.
Lietotājvārds un parole ievadīti pareizi.	Lietotājs tiek pieteikts sistēmā.	Izpildās.
Lietotāja autentificēšanās nav izdevusies, jo savienojumā ar serveri ir problēmas.	Tiek parādīts attiecīgais paziņojums.	Izpildās.

4.2.3. Lietotāja dzēšana

Atsauce uz funkcionālo praīsbu – FP3

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek veikts pieprasījums dzēst lietotāju.	Tiek parādīts uznirstošais logs, kas prasa apstiprinājumu veicamajai darbībai.	Izpildās.
Tiek apstiprināts pieprasījums dzēst lietotāju.	Lietotājs tiek izdzēsts, uznirstošais logs tiek aizvērts un tabulā ar lietotājiem izdzēstais lietotājs vairs neatrodas.	Izpildās.

4.2.4. Lietotāja rediģēšana

Atsauce uz funkcionālo praīsbu – FP4

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek pieprasīta lietotāja rediģēšanas dialogloga atvēršana.	Atveras lietotāja rediģēšanas dialoglogs.	Izpildās.
Ievadītie dati, kas nepieciešami lietotāja rediģēšanai, tiek kļūdaini ievadīti, tiek veikts pieprasījums rediģēt lietotāju.	Tiek iezīmēti kļūdainie lauki un uz ekrāna tiek parādīts uznirstošais logs, kurā parādās informācija par kļūdainajiem laukiem.	Izpildās.
Visi lauki aizpildīti atbilstoši prasībām un tiek veikts pieprasījums lietotāja rediģēšanai.	Uznirstošais logs tiek aizvērts, parādās paziņojums, ka lietotājs ir veiksmīgi rediģēts, un tabulā, kurā uzrādās lietotāji, izmainītais	Izpildās.

	lietotājs tiek attēlots ar izmainīto informāciju.	
Tiek aizpildīts kāds no lietotāja reģistrācijai nepieciešamajiem laukiem.	Ievadot lauku, lauks iezīmējas sarkans, kamēr ievadītā vērtības simbolu skaits ir mazāks par minimālo.	Izpildās.

4.2.5. Apkures sistēmas trīszaru vārstu temperatūras uzstādīšana

Atsauce uz funkcionālo prasību – FP5

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek ievadīta nepareiza vērtība (satur burtus/decimāls skaitlis/tukšs lauks/vērtība, kas ir pārāk liela vai maza), tiek pieprasīta trīszaru vārsta uzstādītās temperatūras maiņa.	Parādās uznirstošais logs ar paziņojumu, ka uzstādāmajai temperatūrai jābūt veselam skaitlim robežās no 0 līdz 75.	Izpildās.
Kā uzstādāmā temperatūra tiek ievadīts vesels skaitlis robežās no 0 līdz 75, tiek pieprasīta trīszaru vārsta uzstādītās temperatūras nomaiņa.	Parādās paziņojums, ka uzstādītā temperatūra ir veiksmīgi nomainīta, trīszaru vārstam tabulā tiek uzrādīta atbilstošā uzstādītā vērtība.	Izpildās.
Savienojuma ar serveri kļūdas dēļ temperatūras nomaiņa neveiksmīga.	Parādās atbilstošais uznirstošais logs.	Izpildās.

4.2.6. Apkures sistēmas izslēgšana

Atsauce uz funkcionālo prasību – FP6

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Visu trīszaru vārstu temperatūras ir uzstādītas uz 0 grādiem.	Apkures sistēmas izslēgšanas pieprasījuma veikšana ir atspējota.	Izpildās.
Visu trīszaru vārstu temperatūras ir uzstādītas uz 0 grādiem un kādam no trīszaru vārstiem temperatūra tiek uzstādīta virs 0 grādiem.	Apkures sistēmas izslēgšana pieprasījuma veikšana tiek iespējota.	Izpildās.
Tiek veikts pieprasījums izslēgt apkures sistēmu.	Visu trīszaru vārstu uzstādītās temperatūras tiek nomainītas uz 0 grādiem.	Izpildās.
Lietotne atvērta divos logos, vienā logā tiek mainīta kāda trīszaru vārsta temperatūra/izslēgta apkures sistēma.	Otrā logā atbilstoši nomainās trīszaru vārsta(u) temperatūra(as) un apkures sistēmas izslēgšana atbilstoši tiek iespējota vai atspējota.	Izpildās.

4.2.7. Ventilācijas regulēšana

Atsauce uz funkcionālo prasību – FP7, FP17

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek mainītas vērtības uz ventilatora režīma regulētāja bez pieprasījuma veikšanas.	Ventilatora režīma diagrammas veida attēlojums mainās atbilstoši vērtībai uz ventilatora režīma regulētāja.	Izpildās.
Tiek veikts pieprasījums nomainīt ventilatora uzstādīto režīmu.	Parādās paziņojums, ka ventilatora uzstādītais režīms veiksmīgi uzstādīts, ventilatora režīms tiek nomainīts.	Izpildās.
Savienojuma ar serveri kļūdas dēļ ventilatora režīms netiek nomainīts.	Parādās attiecīgais paziņojums par to, ka savienojumā ar serveri ir bijusi kļūda un ventilatora režīms netiek nomainīts.	Izpildās.
Tiek veikts pieprasījums nomainīt uzstādīto CO2 līmeni ar nepareizi ievadītiem datiem.	Parādās attiecīgais kļūdas paziņojums.	Izpildās.
Tiek veikts pieprasījums nomainīt uzstādīto CO2 līmeni ar pareizi ievadītiem datiem.	CO2 līmenis tiek nomainīts, tiek attēlots nomainītais CO2 līmenis.	Izpildās.

4.2.8. Vārtu kontrole

Atsauce uz funkcionālo prasību – FP8

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek pieprasīta vārtu aizvēršana.	Tiek iespējota vārtu apturēšanas poga, tiek atspējotas vārtu atvēršanas un aizvēršanas pogas, vārtu statuss tiek parādīts kā „Aizveras”.	Izpildās.
Tiek pieprasīta vārtu atvēršana.	Tiek iespējota vārtu apturēšanas poga, tiek atspējotas vārtu atvēršanas un aizvēršanas pogas, vārtu statuss tiek parādīts kā „Atveras”.	Izpildās.
Vārti ir atvērti.	Nav iespējams pieprasīt vārtu atvēršanu un apturēšanu.	Izpildās.
Vārti ir aizvērti.	Nav iespējams pieprasīt vārtu aizvēršanu un apturēšanu.	Izpildās.

4.2.9. Vārtu apturēšana

Atsauce uz funkcionālo prasību – FP9

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Vārti ir apturēti.	Nav iespējams veikt vārtu apturēšanu.	Izpildās.
Tiek pieprasīta vārtu apturēšana.	Vārtu apturēšanas poga tiek atspējota, vārtu atvēršanas un aizvēršanas pogas tiek iespējotas, vārtu statuss tiek uzrādīts kā „Apturēti”.	Izpildās.

4.2.10. Drošības zonu kontrole

Atsauce uz funkcionālo prasību – FP10

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek veikts pieprasījums ieslēgt signalizāciju drošības zonā.	Drošības zonas signalizācijas ieslēgšanas poga tiek atspējota, drošības zonas signalizācijas izslēgšanas poga tiek iespējota, tiek ieslēgta drošības zonas signalizācija, drošības zonas signalizācijas statusa lauks uzrādās kā „Ieslēgta”.	Izpildās.
Tiek veikts pieprasījums izslēgt signalizāciju drošības zonā.	Drošības zonas signalizācijas izslēgšanas poga tiek atspējota, drošības zonas signalizācijas ieslēgšanas	Izpildās.

	poga tiek iespējota, tiek ieslēgta signalizācija drošības zonā, drošības zonas signalizācijas statusa lauks uzrādās kā „Izslēgta.”	
Veicot pieprasījumu ieslēgt/izslēgt signalizāciju drošības zonā, savienojumā ar serveri rodas kļūda.	Parādās atbilstošais paziņojums, drošības zonas signalizācijas stāvoklis netiek nomainīts.	Izpildās.

4.2.11. Drošības sistēmas kontrole

Atsauce uz funkcionālo prasību – FP11

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Visās drošības zonās signalizācija ir izslēgta.	Signalizācijas sistēmas izslēgšanas poga ir atspējota.	Izpildās.
Visās drošības zonās signalizācija ir ieslēgta.	Signalizācijas sistēmas ieslēgšanas poga ir atspējota.	Izpildās.
Vismaz vienā drošības zonā signalizācijas ir ieslēgta un vismaz vienā drošības zonā signalizācija ir izslēgta.	Signalizācijas ieslēgšanas un izslēgšanas pogas ir iespējotas.	Izpildās.
Tiek veikts pieprasījums ieslēgt signalizācijas sistēmu.	Visās drošības zonās tiek ieslēgta signalizācija, signalizācijas sistēmas ieslēgšanas poga tiek atspējota, signalizācijas sistēmas izslēgšanas poga tiek iespējota.	Izpildās.
Tiek veikts pieprasījums izslēgt signalizācijas sistēmu.	Visās drošības zonās tiek izslēgta signalizācija,	Izpildās.

	signalizācijas sistēmas izslēgšanas poga tiek atspējota, signalizācijas sistēmas ieslēgšanas poga tiek iespējota.	
Veicot pieprasījumu ieslēgt/izslēgt signalizācijas sistēmu, savienojumā ar serveri rodas kļūda.	Parādās atbilstošais paziņojums, drošības sistēmas signalizācijas stāvoklis netiek nomainīts.	Izpildās.

4.2.12. Īpašuma gaismas slēdžu kontrole

Atsauce uz funkcionālo prasību – FP12

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek veikts pieprasījums ieslēgt gaismas avotu.	Gaismas avota ieslēgšanas poga tiek atspējota, gaismas avota izslēgšanas poga tiek iespējota, tiek ieslēgts gaismas avots, gaismas avota statusa lauks uzrādās kā „Ieslēgts”.	Izpildās.
Tiek veikts pieprasījums izslēgt gaismas avotu.	Gaismas avota izslēgšanas poga tiek atspējota, gaismas avota ieslēgšanas poga tiek iespējota, tiek izslēgts gaismas avots, gaismas avota statusa lauks uzrādās kā „Izslēgts”.	Izpildās.

Veicot pieprasījumu ieslēgt/izslēgt gaismas avotu, savienojumā ar serveri rodas kļūda.	Parādās atbilstošais paziņojums, gaismas avota stāvoklis netiek nomainīts.	Izpildās.
--	--	-----------

4.2.13. Visu gaismas avotu kopējā kontrole

Atsauce uz funkcionālo praīsbu – FP13

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Visi gaismas avoti ir ieslēgti.	Visu gaismas avotu kopējā ieslēgšanas poga ir atspējota.	Izpildās.
Visi gaismas avoti ir izslēgti.	Visu gaismas avotu kopējā izslēgšanas poga ir atspējota.	Izpildās.
Vismaz viens gaismas avots ir ieslēgts un vismaz viens gaismas avots ir izslēgts.	Visu gaismas avotu kopējās ieslēgšanas un izslēgšanas pogas ir iespējotas.	Izpildās.
Tiek veikts pieprasījums ieslēgt visus gaismas avotus.	Visi gaismas avoti tiek ieslēgti, visu gaismas avotu kopējā ieslēgšanas poga tiek atspējota, visu gaismas avotu kopējā izslēgšanas poga tiek iespējota.	Izpildās.
Tiek veikts pieprasījums izslēgt visus gaismas avotus.	Visi gaismas avoti tiek izslēgti, visu gaismas avotu kopējā izslēgšanas poga tiek atspējota, visu gaismas avotu kopējā ieslēgšanas poga tiek iespējota.	Izpildās.
Veicot pieprasījumu ieslēgt/izslēgt visus gaismas avotus, savienojumā ar serveri rodas kļūda.	Parādās atbilstošais paziņojums, visu gaismas avotu kopējais stāvoklis netiek nomainīts.	Izpildās.

4.2.14. Sensoru mērījumu pārskata attēlošana

Atsauce uz funkcionālo praīsbu – FP14

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek atvērts sensoru mērījumu pārskata attēlošanas skats.	Zem sensora un laika intervāla ievades tiek attēlots tukšs grafiks ar nenosauktām x un y asīm.	Izpildās.
Nav aizpildīts vismaz viens no aizpildāmajiem laukiem un tiek pieprasīta mērījumu pārskata grafika attēlošana.	Parādās paziņojums par to, ka visiem laukiem ir jābūt aizpildītiem un visi lauki iezīmējas sarkani.	Izpildās.
Tiek aizpildīti visi aizpildāmie lauki un tiek pieprasīta mērījumu pārskata grafika attēlošana.	Tiek uzrādīts grafiks ar visiem izvēlētajā sensora mērījumiem izvēlētajā laika intervālā. X un Y asis tiek nosauktas atbilstoši izvēlētajam sensoram un izvēlētajai valodai. Atkarībā no izvēlētajā sensora tiek rādīts līnijas grafiks vai stabiņu grafiks.	Izpildās.
Tiek aizpildīti visi aizpildāmie lauki un tiek pieprasīta mērījumu pārskata grafika attēlošana, kad noteiktajā laika intervālā izvēlētajam sensoram nav neviena mērījuma.	Tiek parādīts uznirstošais logs ar paziņojumu, ka izvēlētajam sensoram izvēlētajā laika intervālā nav neviena mērījuma.	Izpildās.

4.2.15. Sensoru izvērstās izvēlnes filtrēšana.

Atsauce uz funkcionālo prasību – FP15

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek izvēlēts filtrs.	Atverot izvērsto izvēlni atveras tikai sensori, kuru mērījumi atbilst izvēlētajam filtram.	Izpildās.

4.2.16. Valodas maiņa un uzstādīšana.

Atsauce uz funkcionālo prasību – FP16

Testpiemēra apraksts	Sagaidāmais rezultāts	Rezultāts
Tiek izvēlēta valoda.	Izvēlēta valoda tiek uzstādīta šī brīža sesijā un tiek iestatīta datu bāzē kā lietotājam piesaistītā valoda.	Izpildās.
Pēc valodas izvēles lietotājs izrakstās un atkal autentificējas un piekļūst sistēmai.	Teksti lietotnes skatos tiek uzrādīti lietotāja izvēlētajā valodā.	Izpildās.
Tiek izvēlēta valoda un savienojumā ar serveri ir kļūda.	Šī brīža sesijā valoda tiek nomainīta uz izvēlēto, tiek parādīts paziņojums, ka savienojuma ar serveri kļūdas dēļ lietotājam piesaistītā valoda nav nomainīta.	Izpildās.

5. PROJEKTA ORGANIZĀCIJA

Programmprodukts tika izstrādāts pēc iteratīvās pieejas principa. Sākumā tika definētas definētas produkta pamatprasības un pamatfunkcijas, kas bija pietiekami, lai varētu uzsākt darbu pie programmprodukta izstrādes. Programmprodukta izstrādes laikā tika definētas papildus prasības un sākumā definētās tika konkretizētas.

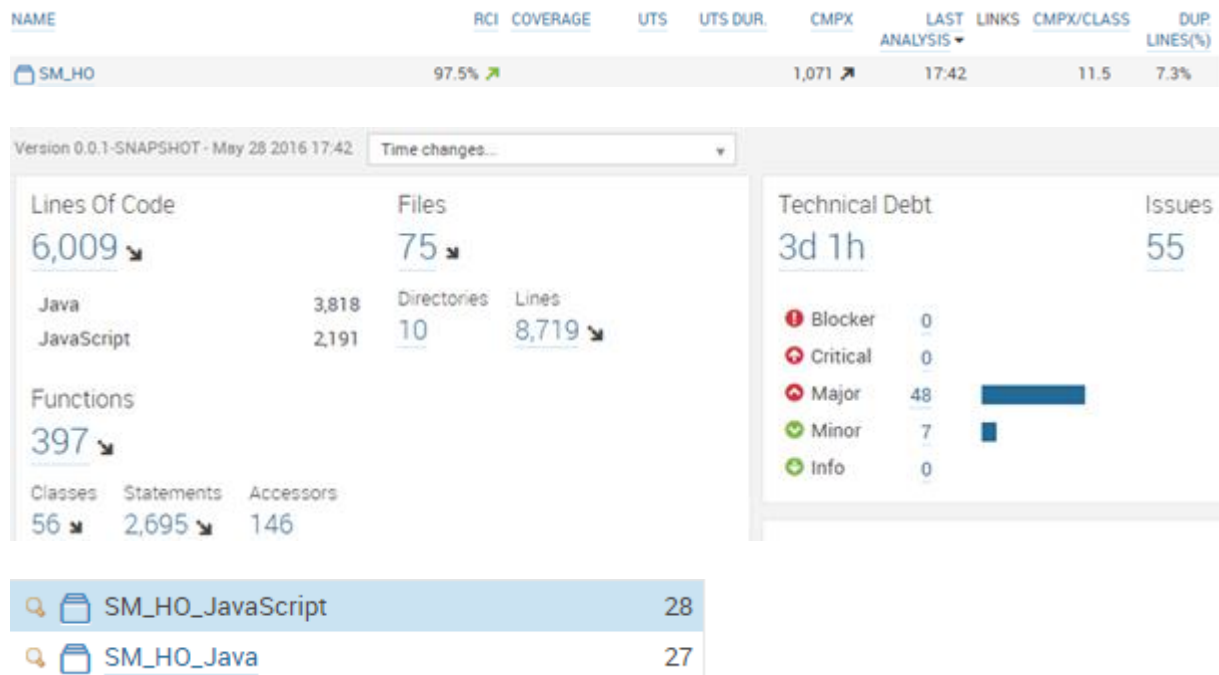
Pārvaldībai tika izmantots Atlassian JIRA rīks. Šis rīks tika izmantots, lai projekta vadītājs definētu uzdevumus un ziņotu par programmprodukta kļūdām, kā arī uzdevumu definēšanai starp komandu, kas piedalījās kopējās “Smart House” lietotnes izstrādē.

Programmprodukta izstrādes versiju kontrole tika nodrošināta ar SVN palīdzību. Lai veiktu programmprodukta testēšanu, tika izmantots nepārtrauktās integrācijas rīks “Jenkins”, kur tika veikti lietotnes būvējumi.

Izstrāde tika veikta Eclipse Luna izstrādes vidē.

6. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu izstrādātā programmaprodukta kvalitāti, tika izmantots SonarQube rīks, kas uzrāda sintakses un formatējuma problēmas, kā arī informāciju par koda rindiņām, kas dublējas un programmkoda sarežģītību. (skat. Attēls 30)

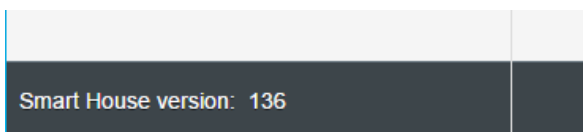


Attēls 30

7. KONFIGURĀCIJAS PĀRVALDĪBA

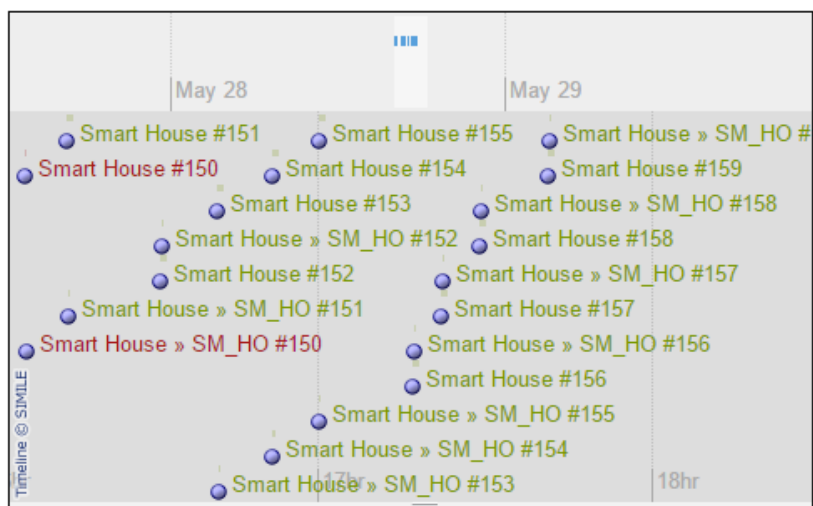
Izstrādājamā programmaprodukta versiju kontrolei tika izmantots *SVN* rīks, kas nodrošina programmaprodukta koda glabāšanu un versiju apskati.

Pēc katra nozīmīga programmaprodukta uzlabojuma tika veikts būvējums, izmantojot nepārtrauktās integrācijas rīku *Jenkins*(skat. Attēls 32), katrs būvējums tika numurēts un lietotāja saskarnes kājēnē tika uzradīts kā programmaprodukta versija(skat. Attēls 31).



Attēls 31

Build History of Jenkins



[Export as plain XML](#)

Build	Time Since ↑	Status	
Smart House » SM_HO #159	2 hr 49 min	stable	
Smart House #159	2 hr 49 min	stable	
Smart House » SM_HO #158	3 hr 1 min	stable	
Smart House #158	3 hr 1 min	stable	

Attēls 32

8. DARBIETILPĪBAS NOVĒRTĒJUMS

8.1. Darbietilpības novērtējums, izmantojot rīku *Startrack*

AS Exigen Services Latvia uzņēmumā projektu pārvaldībai un darbiem patērētā laika uzskaitē tiek izmantots rīks *Startrack*. Darbietilpības novērtējums programmaproduktam “*Gudrā Māja*” lietotāja saskarne tika veidots, izmantojot šajā rīkā ziņotajās atskaites, kas tika veidotas, izstrādājot programmaproduktu. Tiek pieņemts, ka 8 stundas ir viena diena un ka vienā personmēnesī ir 20 dienas.

Veiktais darbs	Dienu skaits
Programmaprodukta izstrāde	48
Jaunās tehnoloģijas <i>SAPUI5</i> iepazīšana	10
Ekrānformu vizuālās daļas izveide	8
Saziņas ar serveri, izmantojot REST tīmekļa servisu, izveide	4
No servera saņemto datu apstrāde un attēlošana	11
Kļūdu apstrāde	7
Izkārtojuma un dizaina, kas reaģē uz ekrāna izmaiņām, izstrāde	4
Lokalizācijas izveide un dinamisko tekstu pielāgošana lokalizācijai	4
Dokumentācijas izstrāde	22
Programmatūras prasību specifikācijas izveide	6
Projektējumu izveide	6
Lietotnes aprakstošā informācija	4
Sākotnējo prasību definēšana un ekrānformu melnrakstu veidošana	3
Testēšana	3
Kopā	70

Kopā programmaprodukta izstrādei tika veltītas 68 dienas, kas pielīdzināms **3,5 personmēnešiem**.

8.2. Darbietilpības novērtējums, izmantojot eksperta metodi

Ņemot vērā, ka SAPUI5 ir ietvars, kas ļauj būtiski samazināt programmatūras koda rindiņu skaitu, darbietilpības aprēķināšanas metodes, kur koda rindiņu skaits ir būtisks faktors (piem., COCOMO), nav piemērotas šī programmaprodukta darbietilpības novērtēšanai.

Konsultējoties ar projekta vadītāju un balstoties uz vadītāja iepriekšējo pieredzi, tika noskaidrots, ka cilvēkam ar pamatzināšanām par Javascript, jQuery un AJAX un bez pieredzes darbā ar SAPUI5 ietvaru šādas lietotāja saskarnes izstrāde aizņem apmēram 3 personmēnešus.

8.3. Izstrādātās koda rindiņas

JavaScript	XML	JSON konfigurācijas fails	Kopā
2191	783	181	3155

9. IZMANTOTĀS TEHNOLOĢIJAS UN RĪKI

Programmprodukta izstrādē izmantotas šādas tehnoloģijas:

- Programmēšanas valodas JavaScript
- CSS
- *jQuery* bibliotēka;
- *AJAX*;
- *XML formāts*;
- *SAPUI5* ietvars;
- Versiju kontroles rīks *SVN*;
- *Jenkins* nepārtrauktās integrācijas rīks;
- *Atlassian JIRA* projektu pārvaldības rīks;
- SonarQube ;
- *Eclipse* programmatūras izstrādes vide;

10. PROGRAMMATŪRAS KODS

10.1. Security moduļa metode "changeZoneMode"

Atsauce uz funkcionālo prasību – FP10

```
/**
 * Generates JSON fragment depending on which button is pressed for
 which zone
 * Performs AJAX request to change zone mode
 * Gets changed zone data from AJAX request and sets it to the zone.
 * Shows error message if zone mode could not be changed
 * @param {object} oEvent - event that triggers this function
 */

changeZoneMode : function (oEvent){
    //Gets security zone's ID and mode that to be set
    var oSource = oEvent.getSource(),
        iId = oSource.data("id"),
        iVal = oSource.getSelectedKey();
    //JSON fragment to be sent
    var oDataToSend = {
        "id":iId,
        "mode":iVal
    };
    //AJAX request that changes the security zone's state
    $.ajax({
        type : "PUT",
        contentType : "application/json",
        url : "rest/securitysystem/securityzone/mode",
        dataType : "json",
        async : false,
        data : JSON.stringify(oDataToSend),
        success : function (oData){
            //The response data is set to corresponding table row
            var aTableData =
this.byId("securityTable").getModel().getProperty('/securityZones');
            for(var i=0; i<aTableData.length; i++){
                var oItem = aTableData[i];
                if(oItem.id === oData.id){
                    aTableData[i] = oData;
                    this.getView().getModel().refresh(true);
                    var oModel = new sap.ui.model.json.JSONModel();

oModel.loadData(util.Utils.getServiceUrl("rest/securitysystem/securitysystem/2"), {}, false, "GET", false, true, {});
                    this.getView().setModel(oModel);
                    //Sets selected button on security systems
                    //control buttons(All On/ All Off)
                    var systemControlMode =
oModel.getProperty('/mode');
                    var systemControl =
sap.ui.getCore().byId(this.createId('systemControl'));
                    if(systemControlMode === 2){
                        systemControl.setSelectedKey("0");}
                }
            }
        }
    });
}
```

```

        else if(systemControlMode === 1){
            systemControl.setSelectedButton("5");
        }
        else if(systemControlMode === 0){
            systemControl.setSelectedKey("2");
        }
        break;
    }
    }
    }.bind(this)
});
},

```

10.2 Administrator moduļa metode "editUser"

Atsauce uz funkcionālo prasību – FP4

```

/**
 * Gets administrator's input and passes it to "validateInput" function
 * If the input is valid, generates a JSON data fragment containing the
input
 * Performs AJAX request to edit a user
 * Shows an error message if user was not edited
 * Shows toast message if the user was edited, closes dialog box
 */
editUser : function() {
    //Gets administrators input
    var oDataName;
    var oDataSurname;
    var oDataPassword;
    var oDataEMail;
    var name = this.getView().byId("firstName").getValue();
    var surname = this.getView().byId("lastName").getValue();
    var confirm = this.getView().byId("confirm").getValue();
    var password = this.getView().byId("password").getValue();
    var email = this.getView().byId("epasts").getValue();
    var oBundle =
this.getComponent().getModel('i18n').getResourceBundle();
    //Calls validateInput function to check if input is valid
    //If not valid, error message is shown
    if(!this.validateInput(name, surname, confirm, password, email,
oBundle)){
        return;
    }
    //Creates password JSON fragment if password was entered
    if(password !== ""){
        oDataPassword = '{"password":' + '"' + password + '"' + '>';
        oDataPassword = JSON.parse(oDataPassword);
    }
    //Creates JSON fragments from administrator's input
    oDataName = '{"firstName":' + '"' + name + '"' + '>';
    oDataSurname = '{"lastName":' + '"' + surname + '"' + '>';
    oDataEMail = '{"epasts":' + '"' + email + '"' + '>';

    ////Puts JSON fragments together

```

```

oDataName = JSON.parse(oDataName);
oDataSurname = JSON.parse(oDataSurname);
oDataEMail = JSON.parse(oDataEMail);

var oDataToSend = $.extend({}, oDataName, oDataSurname,
oDataEMail, oDataPassword);
//Edits user by performing AJAX request on the corresponding user
var oUserData =
sap.ui.getCore().getModel("editUserData").getData();
var oUserDataUsername = oUserData.username;
$.ajax({
type : "PUT",
contentType : "application/json",
url : "rest/users/user/" + oUserDataUsername,
dataType : "json",
data : JSON.stringify(oDataToSend),
success : function(oData) {
//Shows error message if user could not be edited
if(oData.error){
var oMessageType = "Error";
this.showErrorBox(oData.error, oMessageType);
return;}
//Shows toast message if user was edited
sap.m.MessageToast.show("Changes Have Been Saved");
//Sets changed data to corresponding user table row
var parentViewTableModel =
this.getView().getParent().getParent().byId("administratorTable").getModel();
var parentItems = parentViewTableModel.getData();
for(var i =0; i < parentItems.length; i++){
var oItem = parentItems[i];
if(oItem.username === oData.username){
parentItems[i] = oData;
parentViewTableModel.refresh(true);

break;
}
}

//Closes editing dialog
this.getView().getParent().close();
}.bind(this)
});
}

```

10.3. Overview moduļa galvenās metodes

10.3.1. "showChartOverview"

Iegūst lietotāja ievadi, izsauc grafiku sagatavojošās metodes un pēc tam, izmantojot REST servisu, attēlo tajā datus.

```

/**
 * Makes chart that is ready to display graph with sensor records
 * Performs AJAX request to get sensor records
 * Displays sensor records
 * */
showChartOverview : function() {
    //Gets selected time interval and selected sensor
    var inputFrom = this.getView().byId('dateFrom').getDateValue();
    var parsedFrom = Date.parse(inputFrom);
    var inputTo = this.getView().byId('dateTo').getDateValue();

    var parsedTo = Date.parse(inputTo);
    var inputSensor =
this.getView().byId('sensorCombo').getSelectedItem();
    //Error if the input is not valid
    if (inputFrom === "" || inputFrom === null
        || inputSensor === null || inputTo === ""
        || inputTo === null) {
        this.fillFields();
        var oErrorType = 10;
        var oMessageType = "Error";
        this.showErrorBox(oErrorType, oMessageType);
        return;
    }
    //Passes sensor's type to makeChart function
    var inputSensorType = inputSensor.data("type") +
inputSensor.data("address");
    this.makeChart(inputSensorType);

    this.setAxisValues(inputSensorType);

    var inputSensorId = inputSensor.data("id");

    this.fieldsFilled();
    var oDataToSend = {
        "begintimestamp" : parsedFrom,
        "endtimestamp" : parsedTo
    };
    window.oChartmodel = new sap.ui.model.json.JSONModel();
    //Sends AJAX request to get selected sensor's records in the
    //selected time range
    $.ajax({
        type : "PUT",
        contentType : "application/json",
        url : "rest/sensor/sensor/" + inputSensorId,
        dataType : "json",
        async : false,
        data : JSON.stringify(oDataToSend),
        success : function(oData) {
            //Formats data received in format to be displayed
            var oModel = new sap.ui.model.json.JSONModel();
            oModel.setData(oData);
            var aTimeStampData = oModel.getData();
            for (var i = 0; i < aTimeStampData.length; i++) {
                var formattedTimeStamp = aTimeStampData[i].timestamp;
                var milliseconds = new Date(formattedTimeStamp);
                var timeStampFormatted =
                    (milliseconds.getMonth() < 10 ? '0' : '')

```

```

+ (miliSeconds.getMonth()+1) + "/" +
(miliSeconds.getDate() < 10 ? '0': '') +
miliSeconds.getDate() + " " +
(miliSeconds.getHours() < 10 ? '0': '') +
miliSeconds.getHours() + ":" +
+ (miliSeconds.getMinutes() < 10 ? '0': '')
+ miliSeconds.getMinutes()
+ ":" + (miliSeconds.getSeconds() < 10 ? '0': '')
+ miliSeconds.getSeconds();
aTimeStampData[i].timestamp = timeStampFormatted;
}
//Shows message if no data is received
if (oData.length === 0) {
var oErrorType = 13;
var oMessageType = "Information";
this.showErrorBox(oErrorType, oMessageType);
return;
}
window.oChartmodel.setData(aTimeStampData);
}.bind(this)
});
//Sets received data to VizFrame and the records
// are displayed in the VizFrame(chart)
window.oVizFrame.setModel(window.oChartmodel);
}

```

10.3.2. "makeChart"

Sagatavo grafiku konkrēto datu attēlošanai

```

/**
 * Sets dataset to VizFrame control with names and values of axis
 * that are accepted by the VizFrame(chart) and can be displayed
 * In this case the chart is generated dynamically on every request
 * for overview which means that there will only be defined one X
 * axis and one Y axis
 * @param {string}inputSensorType - type of the user's selected sensor
 * */
makeChart : function(inputSensorType) {
    this.getView().byId("idChartContainer").setTitle("");
    window.oVizFrame = this.getView().byId("idoVizFrame");
    //Array that is going to be used for creating a measure element
    var oMeasures = [ {
        group : 1,
        name : inputSensorType,
        value : '{value}'
    }];
    //Defines an empty FlattenedDataset
    var oDataset = new sap.viz.ui5.data.FlattenedDataset(
        {
            data : {
                path : "/"
            }
        });
    //Sets "name" of array to the sensor type's name in current

```

```

        //resource language model
        var oBundle =
this.getComponent().getModel('i18n').getResourceBundle();
        var measure = oMeasures[0].name;
        var name = oBundle.getText("sensor"+ measure);
        oMeasures[0].name = name;
        //Transferring data from array to measure element(defines Y axis)
        var oMeasureElement = new sap.viz.ui5.data.MeasureDefinition();
        oMeasureElement.setGroup(oMeasures[0].group);
        oMeasureElement.setName(oMeasures[0].name);
        oMeasureElement.bindProperty("value", "value");
        //Adds measure element(measure elements define
        //the kind of Y axis that VizFrame will accept)
        oDataset.addMeasure(oMeasureElement);

        //Array that is going to be used for creating a dimension element
        var oDimensions = [ {
            name : 'Time',
            value : "{timestamp}"
        } ];
        //Sets "name" of array to name "Time" in current resource language
model
        //Transferring data from array to dimension element(defines X
axis)
        var dimension = oDimensions[0].name;
        var dimensionName = oBundle.getText("sensor"+ dimension);
        oDimensions[0].name = dimensionName;
        var oDimensionElement = new
sap.viz.ui5.data.DimensionDefinition();
        oDimensionElement.setName(oDimensions[0].name);
        oDimensionElement.bindProperty("value", "timestamp");
        //Adds dimension element(dimension elements define the kind of X
//axis that VizFrame will accept)
        oDataset.addDimension(oDimensionElement);

        window.oVizFrame.setVizProperties({
            plotArea : {
                showGap : true
            }
        });
        //Sets the dataset with defined measures and
//dimensions to VizFrame(chart)
        window.oVizFrame.setDataset(oDataset);
        window.oVizFrame.setVizType('line');
    }
}

```

10.3.3. "setAxisValues"

Sagatavo grafiku konkrēto datu attēlošanai

```

/**
 * Sets FeedItems with names of axis and graph type to column
 * if the selected sensor's records are binary values
 * @param {object}oParameter - type of sensor
 * */

```

```

setAxisValues : function(oParameter) {
    var oBundle =
this.getComponent().getModel('i18n').getResourceBundle();
    //Sets graph type to column for sensors with binary values
    if (oParameter === "P1" || oParameter === "R1") {
        window.oVizFrame.setVizType('column');
        window.oVizFrame.addClass("overviewVizFrameHidden");
    }
    //Sets Y and X axis name to sensor's name and "time" in
    //language resource model
    var values = oBundle.getText("sensor" + oParameter);
    var time = oBundle.getText("Time");
    var YAxisValues = new sap.viz.ui5.controls.common.feeds.FeedItem(
        {
            'uid' : "primaryValues",
            'type' : "Measure",
            'values' : [ values ]
        }
    ),
    XAxisValues = new sap.viz.ui5.controls.common.feeds.FeedItem(
        {
            'uid' : "axisLabels",
            'type' : "Dimension",
            'values' : [ time ]
        }
    );

    window.oVizFrame.removeAllFeeds();
    //Adds FeedItems(which X and which Y axis is going
    //to be shown) to vizframe
    window.oVizFrame.addFeed(YAxisValues);
    window.oVizFrame.addFeed(XAxisValues);
}

```

10.3.4. "generateFilter"

Ģenerē filtru sensoriem, kuri mēra gan temperatūru, gan mitrumu

```

/**
 * Generates a filter for sensors who have two types of measures
 * Filters are generated by accordingly combining oFilter1,
 * oFilter2b, oFilter2c, oFilter3
 * @param {string}oType - determines what type of sensor this is
 * @param {string}oAddress - determines which type of the sensor's
 * @return {object}oFilter123bc - generated filter for the sensor
 * measurement is requested
 * */
generateFilter : function(oType,oAddress) {
    var oFilter1 = new Filter("type","EQ", oType);
    var oFilter2b = new Filter("type","EQ", "B");
    var oFilter3 = new Filter("address", "EQ", oAddress);
    var oFilter23b = new Filter(
        {filters : [ oFilter2b,oFilter3 ],
        and : true
    });
    var oFilter123b = new Filter({
        filters : [ oFilter1,oFilter23b ],

```

```
    or : true
  });
  var oFilter2c = new Filter("type","EQ", "C");
  var oFilter23c = new Filter(
    {
      filters : [ oFilter2c,oFilter3 ],
      and : true
    });
  var oFilter123c = new Filter({
    filters : [ oFilter1,oFilter23c ],
    or : true
  });
  var oFilter123bc = new Filter({
    filters : [ oFilter123b,oFilter123c ],
    or : true
  });
  return oFilter123bc;
}
```

SECINĀJUMI

Projekta mērķis tika sasniegts – lietotnei “Gudrā Māja” izveidota lietotāja saskarne, kas, izmantojot REST tīmekļa servisu, ļauj attālināti kontrolēt ierīces un attēlo informāciju par tām.

Programmprodukta izstrādes laikā tika apgūts līdz šim neizmantots ietvars – SAPUI5, nostiprinātas zināšanas par JavaScript programmēšanas valodu, JSON formāta apstrādi un XML formātu, tika gūta pieredze, izmantojot projektu pārvaldības rīku Atlassian JIRA un versiju kontroles rīku SVN, kā arī SonarQube rīku programmkoda kvalitātes pārbaudei.

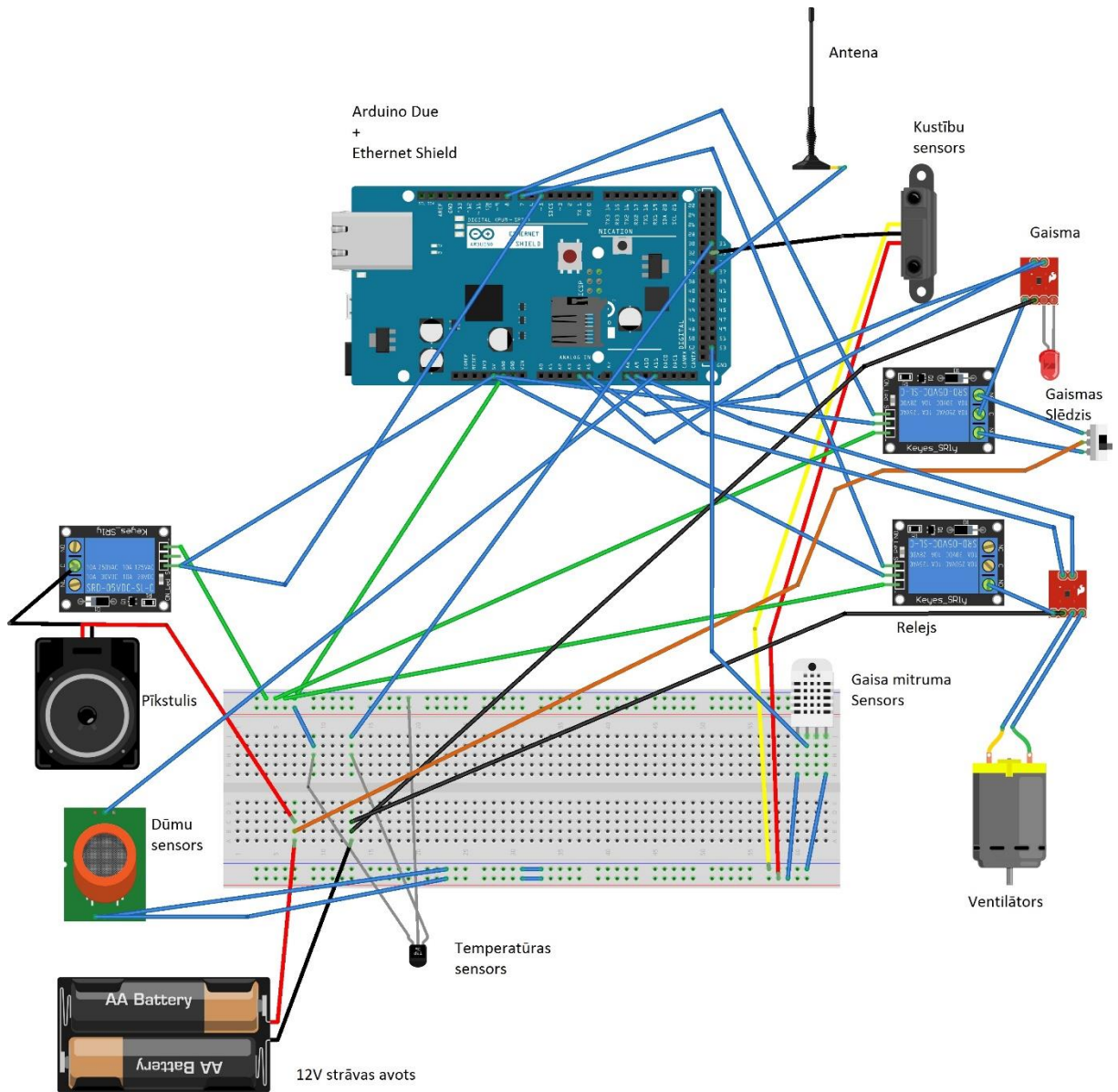
Tika veikti secinājumi, ka koda izstrādi atvieglo koda komentēšana un nosaukumu ar nozīmi piešķiršana mainīgajiem un funkcijām, jo tas ļauj orientēties kodā bez iedziļināšanās.

Tā kā uzsākot programmēšanas darbus, SAPUI5 ietvara zināšanas vēl tika apgūtas, programmproduktā eksistē programmkoda daļas, kuras iespējams optimizēt.

IZMANTOTĀ LITERATŪRA

1. SAPUI5 SDK [tiešsaiste] – [Pārbaudīts 28.05.2016.] Pieejams:
<https://sapui5.netweaver.ondemand.com/sdk/>;
2. Latvijas valsts standarts LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis”;
3. Latvijas valsts standarts LVS 70:1996 “Programmatūras testēšanas dokumentācija”;
4. Latvijas valsts standarts LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai”;
5. JavaScript reference documentation [tiešsaite]. - [Pārbaudīts 28.05.2016] Pieejams:
<https://github.com/ianb/javascript>;
6. Akadēmiskā terminu datu bāze [tiešsaiste] – [Pārbaudīts 28.05.2016] Pieejams:
<http://termini.lza.lv>
7. Daļa no lietotāja saskarnē izmantotajām ikonām [tiešsaiste] – [Pārbaudīts 28.05.2016]
Pieejams: <https://icons8.com/>
8. jQuery API Documentation [tiešsaite] - [Pārbaudīts 26.05.2016]
Pieejams: <http://api.jquery.com/>

PIELIKUMS



fritzing

Attēls 33

Kvalifikācijas darbs „*Guadrā Māja*” lietotāja saskarne 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: *Gustavs Helmutis Felsbergs* _____ .05.2016.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *Dipl.dat. Guntars Šmitiņš* _____ .05.2016.

Recenzents: *M.dat., M.ekon., MBA, Andris Bozis*

Darbs iesniegts 30.05.2015.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2015. prot. Nr. _____

Komisijas sekretārs(-e): _____