

Vidzemes Tehnoloģiju un Dizaina tehnikums

Profesionālās vidējās izglītības programma – Programmēšana

Kvalifikācija – Programmēšanas tehniķis

Izglītības ieguves forma – Klātie

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Tīmekļa lietotne “Your Smart Home”

Autors: IP20 Grupas audzēknis
Krišjānis Klāvs Kantāns

(paraksts)

Cēsis, 2024

ANOTĀCIJA

Viedmājokļa bāzes tīmekļa lietotne “Your Smart Home”, Kantāns, Krišjānis Klāvs.
Kvalifikācijas darbs, 67 lappuses, 97 attēli, 8 tabulas, 20 literatūras avoti, 1 pielikums.

Kvalifikācijas darba mērķis ir izstrādāt mūsdienu prasībām atbilstošu, intuitīvu un noderīgu tīmekļa lietotni, kurā lietotāji var pievienot sensorus un uzglabāt to datus datubāzē, kā arī aplūkot datus detalizētos grafikos ar dažādām datu filtrēšanas iespējām. Ar šīs tīmekļa lietotnes palīdzību lietotāji var noteikt energoneefektīvas telpas, salīdzināt telpu temperatūru un mitrumu savā starpā, sekot līdzi temperatūrai un mitrumam svarīgās vietās, piemēram, siltumnīcās, kā arī aplūkot vēsturiskos datus. Kopumā šī lietotne atvieglo īpašuma uzraudzīšanu attālināti, ļaujot lietotājiem apskatīt reāllaika datus no jebkuras ierīces ar interneta savienojumu.

Darbā aprakstītas izvirzītās prasības rezultātu sasniegšanai, kā arī norādīts, kuras no prasībām ir īstenotas. Svarīgas nodaļas ietver literatūras apskatu un variantu salīdzinājumu, kur tiek salīdzināti citu kompāniju risinājumi ar *Your Smart Home* tīmekļa lietotnes risinājumu. Prasību specifikācijā ir aprakstītas nepieciešamās prasības tīmekļa lietotnes izstrādei. Ir izveidots arī funkcionālais un dinamiskais sistēmas modelis, kas ilustrē lietotāja ceļu caur vietni un datu plūsmu. Lietotāja ceļvedis sniedz instrukcijas un parāda, kā tīmekļa vietne darbojas un kā lietotājs tajā navigē.

Darba mērķis tika sasniegts. *Your Smart Home* tīmekļa lietotnē, lietotāji var pievienot jau gatavus temperatūras un mitruma sensorus, autorizēt tos un glabāt sensoru datus datubāzē, tos datus pēc tam iespējams aplūkot detalizētos grafikos ar statistiku. *Your Smart Home* tīmekļa lietotnes risinājums ar šo funkcionalitāti nodrošina lietotājiem iespēju uzraudzīt temperatūras un mitruma datus tādās vietās kā siltumnīcas, garāžas, mājas telpās u.c.

Tīmekļa lietotnes priekšgalsistēma: <https://home.kantanudarznieciba.lv/>

Tīmekļa lietotnes aizmugurgalsistēma (API): <https://api.kantanudarznieciba.lv/>

GitHub saite ar tīmekļa lietotnes pirmkodu: <https://github.com/xHithy/your-smart-home>

Tīmekļa lietotnes testēšanas profila pieslēgšanās dati:

- lietotājvārds: TestUser123
- parole: PareizaParole123!

ANNOTATION

Smart Home Web Application “Your Smart Home”, Kantāns, Krišjānis Klāvs.
Qualification thesis, 67 pages, 97 images, 8 tables, 20 literature sources, 1 appendix.

The aim of this qualification thesis is to develop a modern, intuitive, and useful web application where users can add sensors and store their data in a database, as well as view the data in detailed charts with various data filtering options. With this web application, users can identify energy-inefficient spaces, compare room temperatures and humidity levels, monitor temperature and humidity in important places such as greenhouses, and view historical data. Overall, this application facilitates remote property monitoring, allowing users to view real-time data from any device with an internet connection.

The thesis describes the requirements set to achieve the results and specifies which of these requirements have been implemented. Important chapters include a literature review and a comparison of alternatives, where solutions from other companies are compared with the solution of the Your Smart Home web application. The requirement specification describes the necessary requirements for the development of the web application. A functional and dynamic system model has also been created, illustrating the user's path through the site and the data flow. The user guide provides instructions and demonstrates how the website works and how the user navigates it.

The goal of the thesis was achieved. In the Your Smart Home web application, users can add ready-made temperature and humidity sensors, authorize them, and store the sensor data in a database, which can then be viewed in detailed charts with statistics. The solution of the Your Smart Home web application with this functionality allows users to monitor temperature and humidity data in places such as greenhouses, garages, home rooms, etc.

Web application frontend: <https://home.kantanudarznieciba.lv/>

Web application backend (API): <https://api.kantanudarznieciba.lv/>

GitHub link with the web application source code: <https://github.com/xHithy/your-smart-home>

Web applications test user credentials:

- username: TestUser123
- password: PareizaParole123!

SATURA RĀDĪTĀJS

Anotācija.....	2
Annotation	3
Satura rādītājs	4
Ievads.....	6
1. Literatūras apskats un variantu salīdzinājums	7
1.1. Amazon Alexa	7
1.2. Google Home.....	8
1.3. Galvenās atšķirības Your Smart Home ar salīdzinātajiem risinājumiem.....	9
2. Tīmekļa lietotnes Prasību specifikācija	10
2.1. Tīmekļa lietotnes Funkcionālās prasības	10
2.2. Tīmekļa lietotnes Nefunkcionālās prasības	17
3. Mērķa sasniegšanas līdzekļu izvēle un pamatojums	19
4. Sistēmas un datu struktūras modelis.....	22
4.1. ER Diagramma	22
4.2. Datubāzes komponentu diagramma	23
4.3. Priekšgalsistēmas komponentu diagramma	23
5. Sistēmas dinamiskais un funkcionālais modelis.....	24
5.1. Lietošanas gadījumu diagramma	24
5.2. Sekvences diagramma	25
5.2.1. Pieslēgšanās tīmekļa vietnei	25
5.2.2. Sensora autorizēšana tīmekļa vietnē.....	26
6. Testēšana.....	27
6.1. Automātiskā testēšana	27
6.1.1. Autorizācijas testi	27
6.1.2. Sensoru darbību testi	28
6.1.3. Viedmājoķļa sekciju testi.....	29
6.2. Tīmekļa lietotnes pārlūkprogrammas pieejamības testi	30

7.	Lietotāja ceļvedis	31
7.1.	Tīmekļa lietotnes atvēršana	31
7.2.	Pieslēgšanās tīmekļa lietotnei	32
7.3.	Navigācijas joslas lietošana	33
7.4.	Sekciju apskate	34
7.5.	Sekcijas pievienošana	35
7.6.	Sekcijas dzēšana	36
7.7.	Sekcijas rediģēšana	37
7.8.	Apakšsekciju apskate	38
7.9.	Apakšsekcijas pievienošana	39
7.10.	Apakšsekcijas rediģēšana	41
7.11.	Apakšsekcijas dzēšana	43
7.12.	Apakšsekcijas piespraušana un atspraušana no sākuma paneļa	44
7.13.	Sensora piesaistīšana apakšsekcijai	47
7.14.	Sensora atsavināšana no apakšsekcijas	48
7.15.	Apakšsekciju datu grafiku laika posma maiņa un paskaidrojumi	50
7.16.	Neautorizētu sensoru apskate un autorizēšana	53
7.17.	Autorizētu sensoru apskate un dzēšana	56
7.18.	Lietotāja izrakstīšanās no tīmekļa lietotnes	59
7.19.	Tīmekļa lietotnes darbību vēstures žurnāla apskate	60
	Secinājumi un priekšlikumi	62
	Izmantotā literatūra	63
	Pielikumi	65

IEVADS

Viedmājokļa, jeb gudrās mājas pakalpojumi ir kļuvuši par būtisku daļu no mūsdienu dzīves, piedāvājot lietotājiem inovatīvas tehnoloģijas, kas atvieglo ikdienas dzīves gaitu. Statistika liecina, ka 2023. gadā pasaulē jau bija apmēram 360 miljoni gudro māju, un šis skaits prognozēts pieaugt līdz 785 miljoniem līdz 2028. gadam (Statista, 2023).

Tās galvenā funkcionalitāte ir nodrošināt lietotājiem iespēju kontrolēt un automatizēt savu mājas vidi, kā arī palīdz atrast energoneefektīvas istabas, izmantojot tehnoloģijas. Šīs mājas ir aprīkotas ar dažādiem sensoriem un savienojamām iekārtām, kas sniedz plašas iespējas efektīvai mājas pārvaldībai. Tomēr, tirgū jau pastāv spēcīgi konkurenti, piemēram, *Amazon Alexa*, *Google Nest* u.c., kuri savās sistēmās uzglabā lietotāju datus.

Šajā kontekstā, mērķis ir izveidot patstāvīgu gudrās mājas bāzi, kur visas sistēmas komponentes atrodas lietotāja īpašumā, tādējādi garantējot lietotāja privātumu un datu drošību. Šis pieejas risinājums atšķiras no esošajiem konkurentiem, nodrošinot lietotājiem kontroli pār saviem datiem un novēršot to nevēlamo nodošanu ārējiem uzņēmumiem.

Mērķis

Kvalifikācijas darba mērķis ir izstrādāt intuitīvu, modernu un datiem drošu viedmājokļa risinājumu bāzi individuāliem klientiem, kas palīdzēs noteikt energoneefektīvas mājokļa telpas, apskatīt vēsturiskos telpu mitruma un temperatūras datus, izveidot savas mājas istabu sarakstu un savienot tos ar temperatūras un mitruma sensoriem, kurus varēs pārvaldīt jebkurā ierīcē ar interneta savienojumu.

Uzdevumi

1. Izstrādāt priekšgalsistēmu (angļu v. *front-end*), kurā lietotāji, varēs pārvaldīt sensoru un to datus ar saskarnes un dažādu grafiku palīdzību.
2. Izveidot datubāzes arhitektūru ER diagrammas veidā, kā arī izstrādāt pašu datubāzi.
3. Izveidot sensoru paraugus un izstrādāt kodu priekš mitruma un temperatūras sensoriem.
4. Izveidot aizmugursistēmu (angļu v. *back-end*), kura datubāzē saglabās sensorus, to parametrus un datus, lietotāja pieslēgšanās datus u.c.
5. Izstrādāt automatizētus testus izmantojot *Laravel* ietvara iebūvēto testēšanas funkciju un veikt manuālu tīmekļa lietotnes testēšanu.
6. Novērst radušās problēmas un veikt labojumus.

1. LITERATŪRAS APSKATS UN VARIANTU SALĪDZINĀJUMS

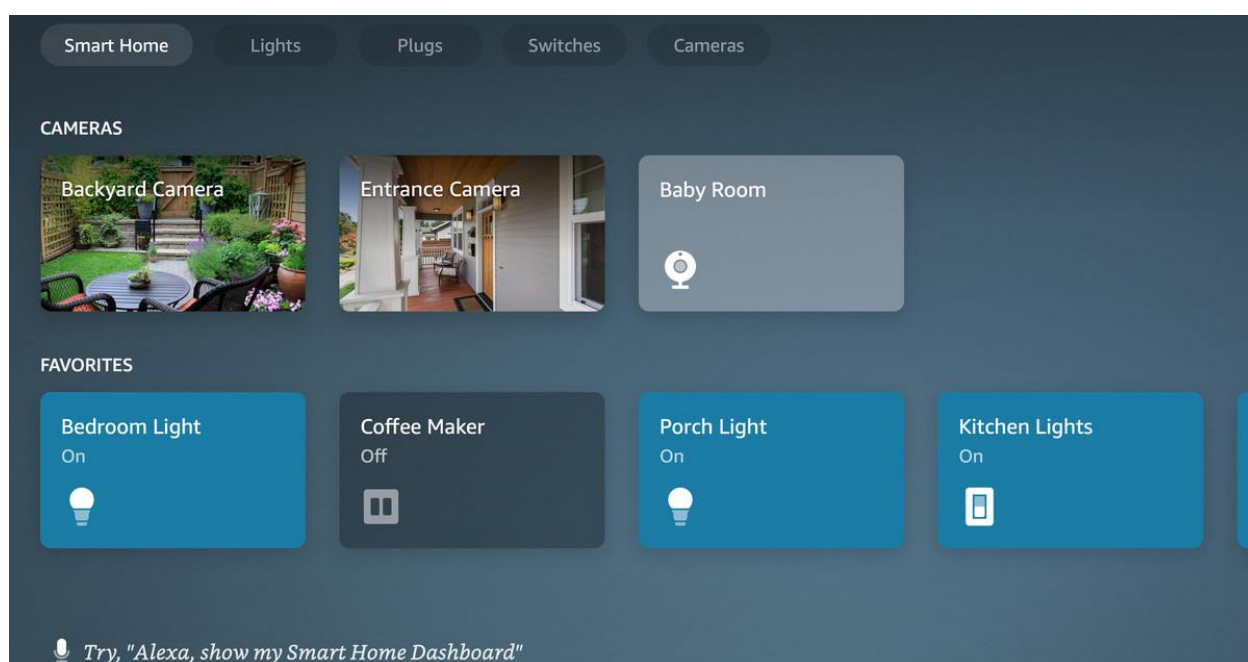
Literatūras apskats un variantu salīdzinājums ir nepieciešams, lai izveidotu plašu zināšanu bāzi par konkrēto tēmu, identificētu galvenos konceptus un metodoloģijas, noteiktu esošos iztrūkumus un nepilnības, kā arī lai veicinātu informētu lēmumu pieņemšanu, novērtējot dažādu risinājumu efektivitāti, ietekmi un riskus, tādējādi uzlabojot projekta kvalitāti un veiksmes iespējas. Šajā nodaļā tiek analizētas alternatīvas *Your Smart Home* risinājumam, tiks noskaidroti galvenie trūkumi un galvenās priekšrocības katrai alternatīvai un tiks salīdzinātas ar *Your Smart Home*.

1.1. AMAZON ALEXA

Amazon Alexa ir viedmājokļu centrālā sistēma, kuru var izmantot, lai pievienotu sensorus un ierīces no daudzu kompāniju klāsta, piemēram, *TP-Link*, *SONOFF*, *Ring*, *IKEA*, *Yeelight* u.c. Dažādo sensoru plašais atbalsts ir viena no galvenajām priekšrocībām, ko piedāvā *Amazon Alexa*, jo praktiski visas viedmājokļa iekārtas ir izstrādātas, ņemot vērā tās savienojamību un darbību ar šo platformu (Layne, 2024).

Paralēli *Amazon Alexa* centrālajai pārvaldības sistēmai, kas var būt *Amazon Echo* formātā - ar balsi kontrolēt un pārvaldīt sensorus un ierīces, *Amazon* arī piedāvā mobilo lietotni, kas ir pieejama gan *Android*, gan *iOS* mobilajās viedierīcēs.

Tās galvenais trūkums ir lietotāju datu privātums. *Amazon Echo*, kas ir balss iekārta, ar ko var kontrolēt visu sensoru darbību, vienmēr ir ieslēgts mikrofons, kas uztver visu, ko lietotājs saka, meklējot atslēgvārdu "Alexa". Kā arī visa centrālā sistēmas datubāze atrodas ārējā serverī un ārpus lietotāja mājokļa. *Amazon Alexa* pārvaldes panela izskats dots 1. attēlā.

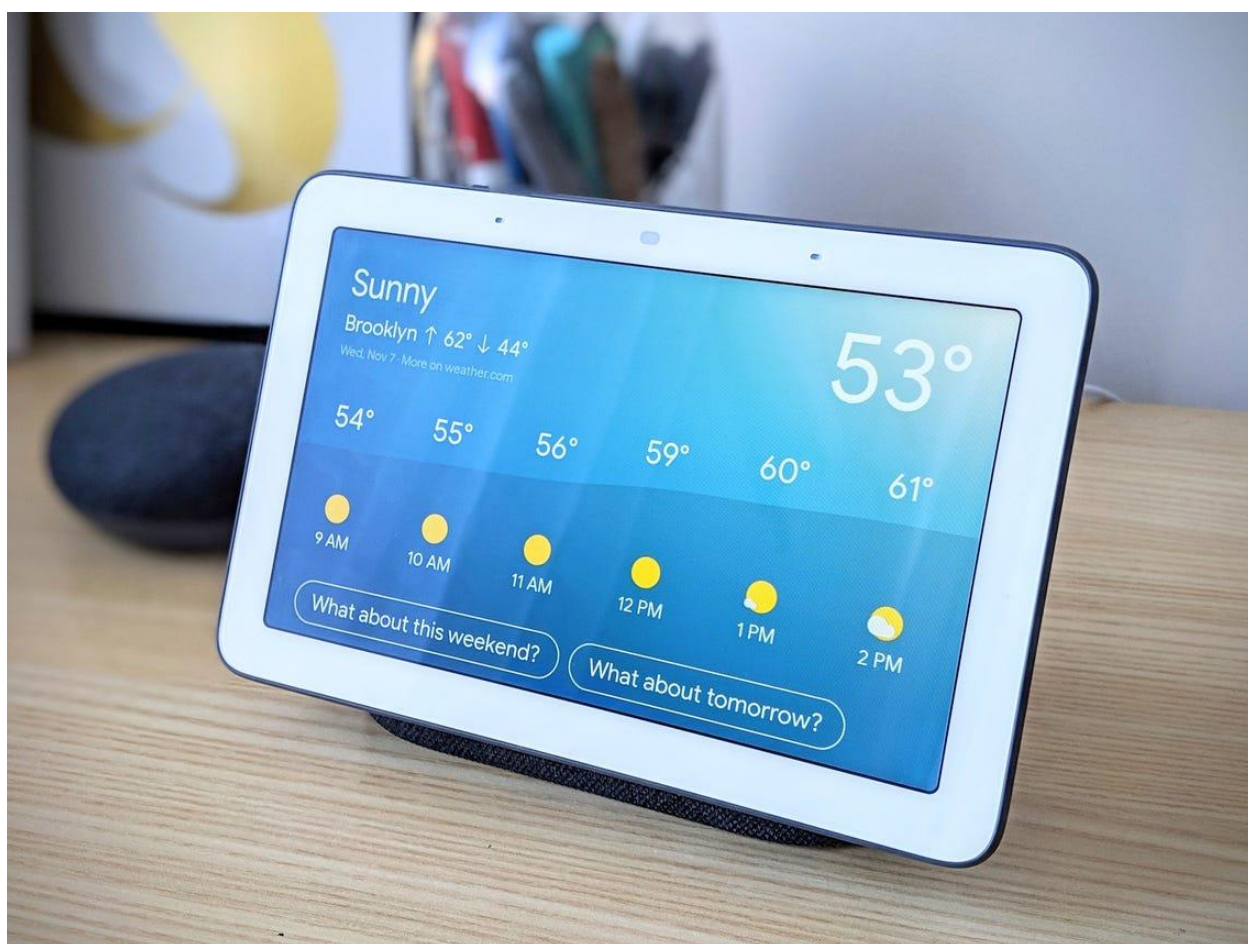


1. attēls. *Google Alexa* galvenais pārvaldes panelis

1.2. GOOGLE HOME

Google Home, ir viedmājokļu centrālā sistēma, kurai klāt nāk arī patstāvīgās *Google* viedmājokļa ierīces, kā piemēram sensori, termostati, centrālās sistēmas ekrāni (*Google Hub*), kameras u.c. *Google Home*, salīdzinājumā ar *Amazon Alexa*, piedāvā daudz plašāku sensoru saderību. Paralēli centrālajai pārvaldības sistēmai, *Google Home* piedāvā arī mobilo lietotni, kas ir pieejama gan *Android*, gan *iOS* mobilajās viedierīcēs. Mobilā lietotne veic tādu pašu funkciju kā standarta centrālā pārvaldības sistēma, kas parasti ir planšetdatora formātā, *Google Hub* planšetdatoru var apskatīt 2. attēlā.

Google Home galvenais trūkums ir tāds pats kā *Amazon Alexa* – datu privātums, lietotāju dati glabājas ārējos serveros, kā arī tas klausās visu laiku priekš atslēgvārda “OK Google” vai “Hey, Google” (BotPenguin, 2023). Šis *Google* produkts ir vislīdzīgākais *Your Smart Home*, jo tiek piedāvāta gan centrālā pārvaldības sistēma, gan patstāvīgi sensori.



2. attēls. *Google Hub* planšetdators ar laikapstākļu lietotni

1.3. GALVENĀS ATŠKIRĪBAS *YOUR SMART HOME* AR SALĪDZINĀJUMIEM

Galvenā atšķirība ir tajā, ka *Your Smart Home* risinājuma klientiem, tiek piedāvāta iespēja uzstādīt serveri ar datubāzi sev vēlamā vietā, tādējādi klientam dodot kontroli pār visiem saviem datiem. Bez tā uzsvars arī tiek likts uz to, ka centrālā pārvaldes sistēmas dizains un izkārtojums būs identisks uz visām mūsdienu ierīcēm, kurām ir pieejams internets, neieskaitot viedpulksteņus.

Galvenais trūkums ar salīdzinājumiem būs tas, ka nebūs ārēju kompāniju viedās mājas ierīču atbalsts, tādēļ *Your Smart Home*, piedāvās jau gatavas un iztestētas ierīces. Vēl trūkums ir tajā, ka visas ierīces un centrālā pārvaldes sistēma komunicē savā starpā ar Wi-Fi savienojumu, tādēļ, interneta savienojuma trūkuma gadījumā, visa sistēma tiek apstādināta.

2. TĪMEKĻA LIETOTNES PRASĪBU SPECIFIKĀCIJA

Tīmekļa lietotnei jānodrošina datu ievade, apstrāde un saglabāšana, reāllaika datu saņemšana no sensoriem ar API starpniecību, lai tā veiksmīgi funkcionētu. Tai jābūt intuitīvai un modernai, izmantojot modernākās WEB programmēšanas tehnoloģijas un valodu ietvarus, kā *ReactJS* un *Laravel* (fastfwd, 2022) (Oleksandr Hutsulyak, 2023).

2.1. TĪMEKĻA LIETOTNES FUNKCIONĀLĀS PRASĪBAS

Funkcionālās prasības ir programmatūras vai sistēmas specifikācijas, kas apraksta, kādas darbības vai funkcijas jāveic sistēmai, lai tā sasniegtu savus mērķus un nodrošinātu noteiktu funkcionalitāti. Funkcionālās prasības fokusējas uz konkrētām darbībām, ko sistēma vai programmatūra ir spējīga izpildīt, kā arī uz to rezultātiem (Ela Mazurkiewicz, 2023).

Funkcionāla Prasība: lietotājam tiek dota iespēja pieslēgties savam personīgajam kontam ko izveidojis administrators sistēmas uzstādīšanas laikā

Izstrādājamā vienība, jeb User Story Man kā sistēmas lietotājam ir nepieciešams piekļūt tīmekļa lietotnei ar sev piešķirtajiem piekļuves datiem

Scenārijs Veiksmīgas pieslēgšanās scenārijs jeb lietotājs ir veiksmīgi autentificējies

Ja lietotājs ir ievadījis lietotājvārdu kas ir reģistrēts datubāzē

Un ja lietotājvārds nepārsniedz 20 rakstzīmes un nav mazāk par 3 rakstzīmēm

Un ja lietotājs ir ievadījis eksistējošu paroli kas ir saistīta ar ievadīto lietotājvārdu

Tad lietotājs tiek pāradresēts uz galveno pārvaldes paneli

Scenārijs lietotājs ir ievadījis lietotājvārdu kas pārsniedz 20 rakstzīmes vai mazāk par 3 rakstzīmēm

Ja lietotājs ir ievadījis neatbilstoši noformētu lietotājvārdu

Lietotājvārds nedrīkst pārsniegt 20 rakstzīmes

Lietotājvārds nedrīkst būt mazāks par 3 rakstzīmēm

Lietotājvārda ievadlauks nedrīkst būt tukšs

Tad parādās modālais logs, kas paskaidro to ka lietotājvārds ir nepareizi noformatēts

Funkcionālā prasība lietotājam tiek dota iespēja apskatīt sensorus kas mēģina autorizēties

Izstrādājamā vienība, jeb User Story man kā sistēmas lietotājam ir nepieciešams redzēt sensorus kas šobrīd mēģina autorizēties ar serveri

Konteksts

Kad lietotājs ir autorizējies

Un kad ir atvēris sadaļu "Settings" -> "Unauthorized Sensors"

Scenārijs veiksmīga iznākuma scenārijs

Ja šobrīd vismaz viens sensors mēģina savienoties ar viedmājokļa serveri

Un ja sensora autorizācijas pieprasījums nav vecāks par 35 sekundēm

Un ja pieprasījuma saturā no sensora ir identifikatora žetons 30 rakstzīmju garumā (Tikai burti un cipari)

Tad sarakstā tiek pievienoti sensori kas šobrīd mēģina veikt savienojumu

Scenārijs nav sensoru, kas mēģina savienoties

Ja šobrīd nav sensoru, kuru pēdējais pieprasījums nav vecāks par 35 sekundēm

Tad sensoru vietā parādīsies paziņojums, par to, ka nav atrasts neviens sensors

Funkcionālā prasība lietotājam tiek dota iespēja autorizēt sensorus, kas mēģina savienoties

Izstrādājamā vienība, jeb User Story man kā sistēmas lietotājam ir nepieciešams autorizēt sensoru kas mēģina savienoties ar serveri

Konteksts

Kad lietotājs ir autorizējies

Un kad ir atvēris sadaļu "Settings" -> "Unauthorized Sensors"

Un kad ir nospiesta poga "Authorize"

Scenārijs veiksmīgs sensora apstiprināšanas scenārijs

Ja sensora pēdējais autorizācijas pieprasījums nav vecāks par 35 sekundēm

Tad konkrētais sensors tiek pievienots autorizēto sensoru sarakstā un turpmāk sūta datus uz serveri, kas tos glabās datubāzē

Funkcionālā prasība lietotājam tiek dota iespēja rediģēt autorizētā sensora nosaukumu

Izstrādājamā vienība, jeb User Story man kā sistēmas lietotājam ir nepieciešams rediģēt sensora nosaukumu

Konteksts

Kad lietotājs ir autorizējies

Un kad ir atvēris sadaļu "Settings" -> "Authorized Sensors"

Un kad kursora nospiests uz sensora nosaukuma

Tad nosaukums parādās ievadlaukā kuru var rediģēt

Scenārijs veiksmīga sensora nosaukuma rediģēšana

Ja ievadītais nosaukums ir garāks par 2 rakstzīmēm

Un ja ievadītais nosaukums nav garāks par 20 rakstzīmēm

Un ja no ievadlauka pazūd fokuss

Tad nosaukums tiek nomainīts saskarne un datubāzē atsevišķi, lai netiek lieki noslogots serveris ar pieprasījumiem

Scenārijs lietotājs mēģina nomainīt sensora nosaukumu uz to pašu kāds bija

Ja ievadītais nosaukums ir tāds pats kāds tas bija pirms tam

Un ja ievadītais nosaukums nav garāks par 20 rakstzīmēm

Un ja no ievadlauka pazūd fokuss

Tad pieprasījums serverim netiek veikts un nekādas darbības netiek veiktas

Scenārijs ievadlauks ir nepareizi noformatēts

Ja sensora nosaukums pārsniedz 20 rakstzīmes vai ir mazāks par 2 rakstzīmēm

Tad uz lietotāja ekrāna labajā augšējā stūrī parādās modālais logs ar paziņojumu, ka ievadlauks ir nepareizi noformatēts

Funkcionālā prasība lietotājam tiek dota iespēja izveidot jaunu sekciju virtuālās mājas izkārtojumā

Dizaina paraugs

Izstrādājamā vienība, jeb User Story man kā sistēmas lietotājam, vajag izveidot sekciju, lai turpmāk tajās varētu pievienot apakšsekcijas

Konteksts

Kad lietotājs ir autorizējies

Un kad ir atvēris sadaļu "House"

Un kad nospiesta poga "New Section"

Tad uz ekrāna parādas modālais logs "Create a section" ar ievadlaukiem

Scenārijs veiksmīga viedmājokļa sekcijas pievienošana

Ja ievadlauks "Section name" nepārsniedz 55 rakstzīmes un nav mazāks par 2 rakstzīmēm

Un ja tiek nospiesta poga "Finish"

Tad uz lietotāja ekrāna labajā augšējā stūrī parādīsies zaļš modālais paziņojums ar informāciju, ka ir veiksmīgi pievienota sekcija, un ievadlauku modālais logs aizvērsies automātiski, kā arī sarakstā bez tīmekļa vietnes pārlādes parādīsies tikko pievienotā sekcija

Scenārijs "Section name" ievadlauks ir nepareizi noformēts

Ja ievadlauks "Section name" pārsniedz 55 rakstzīmes vai ir mazāks par 2 rakstzīmēm

Un tiek nospiesta poga "Finish"

Tad zem "Section name" ievadlauks būs apzīmēts sarkanā krāsā un zem tā būs sarkans paskaidrojošais teksts ar informāciju par to kāda kļūda

Funkcionālā prasība lietotājam tiek dota iespēja pievienot jaunu apakšsekciju kādai jau esošajai sekcijai virtuālās mājas izkārtojumā

Izstrādājamā vienība, jeb User Story man kā sistēmas lietotājam, vajag pievienot apakšsekciju kādai jau esošai sekcijai

Konteksts

Kad lietotājs ir autorizējies

Un kad ir atvēris sadaļu "House"

Un kad ir atvērta kāda sekcija

Tad uz ekrāna parādās modālais logs "Create a room" ar ievadlaukiem

Scenārijs veiksmīga viedmājokļa apakšsekcijas pievienošana

Ja ievadlauks "Room name" nepārsniedz 55 rakstzīmes un nav mazāks par 2 rakstzīmēm

Un ja izvēles laukā "Category" ir izvēlēta kāda no pieejamajām kategorijām

Un ja tiek nospiesta poga "Finish"

Tad uz lietotāja ekrāna labajā augšējā stūrī parādīsies zaļš modālais paziņojums ar informāciju, ka ir veiksmīgi pievienota apakšsekcija, un ievadlauku modālais logs aizvērsies automātiski, kā arī sarakstā bez tīmekļa vietnes pārlādes parādīsies tikko pievienotā apakšsekcija

Scenārijs "Room name" ievadlauks ir nepareizi noformēts

Ja ievadlauks "Room name" pārsniedz 55 rakstzīmes vai ir mazāks par 2 rakstzīmēm

Un tiek nospiesta poga "Finish"

Tad "Room name" ievadlauks būs apzīmēts sarkanā krāsā un zem tā būs sarkans paskaidrojošais teksts ar informāciju par to kāda kļūda

Scenārijs "Category" izvēles lauks ir nepareizi noformēts

Ja izvēles laukā "Category" nav izvēlēta neviena kategorija

Un tiek nospiesta poga "Finish"

Tad "Category" izvēles lauks būs apzīmēts sarkanā krāsā un zem tā būs sarkans paskaidrojošais teksts ar informāciju par to kāda kļūda

Funkcionālā prasība lietotājam tiek dota iespēja izdzēst visus sakrātos datus no sensoriem

Izstrādājamā vienība, jeb User Story man kā sistēmas lietotājam, vajag izdzēst visus datus no sensoriem

Konteksts

Kad lietotājs ir autorizējies

Un kad ir atvēris sadaļu "Settings" -> "Wipe Data"

Tad uz ekrāna parādās modālais logs "Wipe data"

Scenārijs veiksmīga datu izdzēšana

Ja ievadlauks "Password confirmation" satur pareizu paroli kas atbilst aktīvajam lietotājam

Un tiek nospiesta poga "Wipe data"

Tad visi dati tiek izdzēsti, uz lietotāja ekrāna parādās zaļš modālais logs ar paziņojumu, ka dati ir veiksmīgi izdzēsti, "Wipe data" modālais logs automātiski aizveras

Scenārijs "Password confirmation" ievadlauks nepareizi noformēts

Ja ievadlauks ir nepareizi noformēts

Ievadlauks ir tukšs

Ievadlauka parole nesakrīt ar aktīvā lietotāja paroli

Un tiek nospiesta poga "Wipe data"

Tad "Password confirmation" ievadlauks būs apzīmēts sarkanā krāsā un zem tā būs sarkans paskaidrojošais teksts ar informāciju par to kāda kļūda

Funkcionālā prasība lietotājam tiek dota iespēja dzēst sekciju no viedmājokļa sekciju saraksta

Izstrādājamā vienība, jeb User Story man kā sistēmas lietotājam, vajag sekciju no sekciju saraksta

Konteksts

Kad lietotājs ir autorizējies

Un kad ir atvēris sadaļu "Home"

Un kad ir nospiesta trīspunktu ikona uz kādas no pieejamām sekcijām

Un kad ir nospiesta poga "Delete"

Tad uz ekrāna parādās modālais logs "Confirm deletion"

Scenārijs veiksmīga sekcijas dzēšana

Ja nospiesta poga "Delete" atvērtajā modālajā logā

Tad uz lietotāja ekrāna labajā augšējā stūrī parādās zaļš modālais logs ar paziņojumu, ka sekcija veiksmīgi izdzēsta, atvērtais modālais logs automātiski aizveras un sekciju sarakstā bez lapas pārlādes tiek atjaunots sekciju saraksts

Funkcionālā prasība lietotājam tiek dota iespēja dzēst apakšsekciju no viedmājokļa apakšsekciju saraksta

Izstrādājamā vienība, jeb User Story man kā sistēmas lietotājam, vajag apakšsekciju no apakšsekciju saraksta

Konteksts

Kad lietotājs ir autorizējies

Un kad ir atvēris sadaļu "Home" un atvēris kādu pieejamo sekciju

Un kad ir nospiesta trīspunktu ikona uz kādas no pieejamām apakšsekcijām

Un kad ir nospiesta poga "Delete"

Tad uz ekrāna parādās modālais logs "Confirm deletion"

Scenārijs veiksmīga apakšsekcijas dzēšana

Ja nospiesta poga "Delete" atvērtajā modālajā logā

Tad uz lietotāja ekrāna labajā augšējā stūrī parādās zaļš modālais logs ar paziņojumu, ka apakšsekcija veiksmīgi izdzēsta, atvērtais modālais logs automātiski aizveras un apakšsekciju sarakstā bez lapas pārlādes tiek atjaunots apakšsekciju saraksts

2.2. TĪMEKĻA LIETOTNES NEFUNKCIONĀLĀS PRASĪBAS

Nefunkcionālās prasības ir programmatūras vai sistēmas īpašības un kritēriji, kas neattiecas uz konkrētiem funkcionalitātes aspektiem, bet gan uz sistēmas vispārējo veiktspēju, drošību, uzticamību, pielāgojamību un citiem nemateriāliem jautājumiem. Tās nosaka, kā sistēmai jādarbojas, neietilpstot konkrētās funkcijās vai darbībās (Ela Mazurkiewicz, 2023) (*Your Smart Home* nefunkcionālās prasības skatīt 1. tabulā).

1. tabula

Nefunkcionālās prasības

Nr.	Nosaukums	Apraksts
1.	Reaģējoša tīmekļa lietotne visās tīmekļa lietotnes lapās.	Neatkarīgi no ekrāna izmēra (līdz 320px platumam) lietotnei vienmēr būs nodrošināta lietojamība uz jebkura ekrāna izmēra.
2.	Valoda.	Tīmekļa lietotne ir pieejama angļu valodā.
3.	Sensoru datu attēlojums.	Katram pievienotajam sensoram vēsturiskie dati ir apskatāmi viegli saprotamā un vizuāli pievilcīgā grafikā.
4.	Tīmekļa lietotnes veiktspēja.	Tīmekļa vietnes veiktspēja tiek nodrošināta ar sistēmas stresa testa pārbaudēm. Veiktspēja arī tiks nodrošināta ar optimizētu dažādu sensoru datu saglabāšanu servera kešatmiņā, lai netiktu veikti lieki un laikietilpīgi vaicājumi datubāzei.
5.	Tīmekļa vietnes drošības nodrošināšana.	Tīmekļa vietnes priekšgalsistēma tiks balstīta uz TypeScript, ar to nodrošinot lietotnes drošību lai izvairītos no neparedzamiem un nestandarta scenārijiem. Lietotāju autorizācija tiks balstīta uz žetoniem, kas pēc noteikta laika būs jāatjauno ar paroles ievadīšanu. Sensori nespēs savienoties ar viedmājokļa centrālo pārvaldes sistēmu bez lietotāja apstiprināšanas.
6.	Tīmekļa vietnes starp platformu saderība	Tīmekļa vietne būs pieejama uz jaunākajām <i>Safari, Microsoft Edge, Google Chrome</i> un

		<i>Firefox</i> pārlūkprogrammām. Kā arī tā būs pieejama uz <i>iOS</i> , <i>Android</i> , <i>MacOS</i> un <i>Windows</i> operētājsistēmām.
7.	Tīmekļa lietotnes izkārtojums	Tīmekļa lietotnes izkārtojumam jābūt intuitīvam uz jebkuras ierīces virs 320 pikseļu platuma dimensijām.

3. MĒRĶA SASNIEGŠANAS LĪDZEKĻU IZVĒLE UN PAMATOJUMS

HTML - Hiperteksta iezīmēšanas valoda (*HTML*), kas ir paredzēta priekš tīmekļa lappušu un citu pārlūkprogrammu attēlojamās informācijas glabāšanai. Bez šīs valodas nespētu veidot tīmekļa vietnes, jo tā definē un ievieto elementus, kas satur galveno informāciju par vietni (Astari S, 2023).

JavaScript - programmēšanas valoda, kas ir būtiska tīmekļa izstrādei. Tā ļauj veidot dinamiskas un interaktīvas tīmekļa lapas, piedāvājot iespēju manipulēt *HTML* un *CSS*, reaģēt uz lietotāju darbībām un veikt daudzus citus uzdevumus, kas uzlabo lietotāju pieredzi. Bez *JavaScript* mūsdienu tīmekļa vietnes būtu statiskas un ierobežotas savā funkcionalitātē. Šajā tīmekļa lietotnē *JavaScript* es izmantošu netieši, caur *ReactJS* ietvaru, atsevišķos gadījumos izmantojot *JavaScript* *DOM* manipulēšanas funkcijas (Glossary, 2022).

ReactJS - *JavaScript* bibliotēka, kas izstrādāta, lai veidotu lietotnes. *ReactJS* ļauj izstrādātājiem viegli izveidot sarežģītas lietotnes, izmantojot komponentu bāzētu pieeju. Šis ietvars piedāvā efektīvu un elastīgu veidu, kā pārvaldīt tīmekļa lietotnes stāvokli un lietotāju darbības, ļaujot veidot ātras un reaģējošas tīmekļa lietotnes. Bez *ReactJS* tīmekļa izstrāde būtu sarežģītāka un laikietilpīgāka, jo šis ietvars ievērojami vienkāršo un paātrina izstrādes procesu. Šajā tīmekļa lietotnē es izmantošu *ReactJS* lai izveidotu dinamisku un reaģējošu priekšgalsistēmu.

PHP - servera puses skriptu valoda, kas ir izstrādāta tīmekļu vietnes izveidei un dinamiskai darbībai, tā atbalsta formu apstrādi, datu bāzes pieprasījumus un citas funkcijas. Priekš šīs tīmekļu vietnes es *PHP* izmantoju netiešā veidā, caur *Laravel* ietvaru, kur es apstrādāju datubāzes datus, izvadot tos, dzēšot, rediģējot un veidojot dažādus *SQL* saucienus (Rory Toal, 2022).

Laravel - *PHP* ietvars, kas paredzēts, lai atvieglotu tīmekļa lietojumprogrammu izstrādi. Izstrādāts ar mērķi uzlabot izstrādātāju produktivitāti, *Laravel* piedāvā elegantu sintaksi, daudz iebūvētu funkcionalitāšu un rīku, kas palīdz veidot uzticamas un uzturamas lietojumprogrammas. Šajā tīmekļa lietotnē *Laravel* ietvaru lietošu, lai atvieglotā pierakstā un failu struktūrā izveidotu aizmugursistēmas struktūru un veiktu darbības ar datubāzēm.

MySQL - atvērta pirmkoda relāciju datubāžu pārvaldības sistēma (RDBMS), kas balstīta uz *SQL* (Structured Query Language). Tā ir īpaši piemērota lielu un dinamisku tīmekļa lietojumprogrammu datu glabāšanai un pārvaldīšanai (Kashyap Vyas, 2023).

Tailwind - *CSS* rīku kopa, kas palīdz ātrāk un efektīvāk izveidot stilu un izkārtojumu tīmekļu vietnēm. Šis rīku komplekts iekļauj lielu skaitu klases, kas atvieglo izkārtojumu un stilu definēšanu. Tas piedāvā arī saderīgus un pielāgojamus iepriekš definētus dizainus un krāsu paleti, kas palīdz ātri un vienkārši pielāgot tīmekļa lietotnes izskatu (Thomas Ryu, 2023).

Arduino valoda - programmēšanas valoda, kas balstīta uz *C/C++* un ir paredzēta *Arduino* mikrokontrolieru programmēšanai. Tā nodrošina vienkāršu un pieejamu veidu, kā izstrādāt un ieviest dažādus elektronikas projektus, izmantojot sensorus, aktuatorus un citas komponentes. Šajā tīmekļa lietotnē *Arduino* valodu izmantošu lai rakstītu mikrokontrolieru loģiku un lai savienotu to ar tīmekļa lietotnes serveri.

ESP8266 - zemas cenas *Wi-Fi* mikrokontrolieris, kas ir plaši izmantots interneta lietu (IoT) projektos. Tas nodrošina iespēju savienot dažādus sensorus un ierīces ar internetu, ļaujot nosūtīt sensora datus uz serveri vai mākoņpakalpojumu. Šajā tīmekļa lietotnē *ESP8266* mikrokontrolieri izmantošu kā sensoru bāzi, uz kuru sensors nodos datus un tad tālāk mikrokontrolieris tos nodos uz serveri kur tie tiks apstrādāti.

PHPStorm - komerciāla, pilnvērtīga integrētā izstrādes vide (IDE), kas ir speciāli izstrādāta *PHP* izstrādātājiem. Tā piedāvā plašu funkcionalitāti, kas uzlabo izstrādes procesu, ieskaitot gudru koda rediģēšanu, automātisko pabeigšanu, koda pārbaudi un kļūdu labošanas rīkus. *PhpStorm* atbalsta arī citas tīmekļa tehnoloģijas, piemēram, *HTML*, *CSS*, *JavaScript* un *SQL*, ļaujot izstrādātājiem strādāt ar visu projektu vienā vidē.

Docker - atvērtā pirmkoda platforma, kas ļauj izstrādātājiem automatizēt izvietošanu, mērogošanu un pārvaldību konteineru veidā. Konteineri ir viegli, pārnēsājami izpildlaika vidējie līdzekļi, kas ietver visu, kas nepieciešams lietojumprogrammas darbināšanai, piemēram, kodu, bibliotēkas un atkarības. *Docker* nodrošina konsekventu izpildlaiku dažādās izstrādes un ražošanas vidēs, neatkarīgi no pamatā esošās infrastruktūras. Šajā projektā *Docker* izmantošu lai izstrādes laikā izveidotu lokālu serveri (Hao, 2023).

WebSockets - komunikācijas protokols, kas nodrošina pilnduplekša (divvirzienu) sakaru kanālu starp tīmekļa pārlūkprogrammu un serveri pār vienu TCP savienojumu. Tas ir izstrādāts, lai atvieglotu reāllaika datus apmaiņu tīmekļa lietojumprogrammās, padarot to par ideālu risinājumu čata lietotnēm, spēlēm, tiešraides atjauninājumiem un citām lietotnēm, kurām nepieciešama ātra un nepārtraukta saziņa. Šajā tīmekļa lietotnē *WebSockets* izmantošu lai būtu reāllaika datu apmaiņa starp *Laravel* serveri, *ESP8266* mikrokontrolieri un lietotnes priekšgalsistēmu, tādējādi piegādājat un attēlojot sensora datus pēc iespējas ātrāk (Alex Diaconu, 2023).

GitHub ir populāra platforma, kas nodrošina versiju kontroles un sadarbības iespējas programmatūras izstrādē, izmantojot Git. Tā ir ideāla vieta projektu mitināšanai, koda pārvaldībai un sadarbībai starp izstrādātājiem. *GitHub* piedāvā funkcionalitāti, kas atvieglo kodu pārskatīšanu, problēmu izsekošanu un dokumentāciju, padarot to par būtisku rīku mūsdienu programmatūras izstrādes procesos. Šajā projektā uzturēšanai un automatizācijai *GitHub* nodrošina *GitHub Actions* – CI/CD (Continuous Integration/Continuous Deployment) rīku, kas ļauj automatizēt dažādas

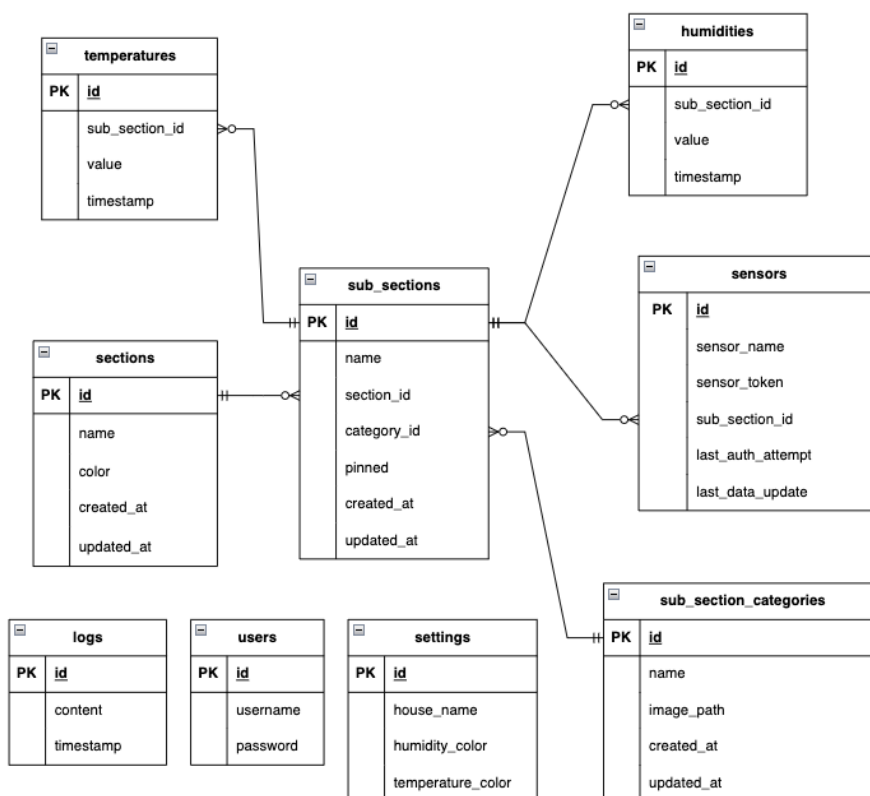
darbplūsmas tieši no *GitHub* repozitorija. Ar *GitHub Actions* palīdzību var veidot, testēt un izvietot kodu automātiski, balstoties uz noteiktiem notikumiem, piemēram, izmaiņām kodā vai izlaidēs versijām. Konkrēti pielietoju to, ka uz servera tiek uzlikti repozitorijā atjaunotie dati – tādējādi piegādājot serverim jaunāko kodu kuru varēs redzēt uz publiskā domēna (dev.to, 2023).

4. SISTĒMAS UN DATU STRUKTŪRAS MODELIS

Sistēmas un datu struktūras modelis ir būtisks datu modeļa veids, kas attēlo sistēmas loģisko un fizisko struktūru, definējot datu elementus, to attiecības un savstarpējo saistību. Šis modelis ietver detalizētu datu shēmu izstrādi, tabulu un to laukumu definīcijas, kā arī datu plūsmas diagrammas, kas palīdz izprast, kā informācija tiek glabāta, pārvaldīta un apstrādāta sistēmā. Šis modelis ir svarīgs, lai nodrošinātu datu integritāti, konsekveni un efektīvu piekļuvi datiem, kā arī lai veicinātu sistēmas izstrādi, kas ir pielāgota lietotāju vajadzībām un funkcionālajām prasībām.

4.1. ER DIAGRAMMA

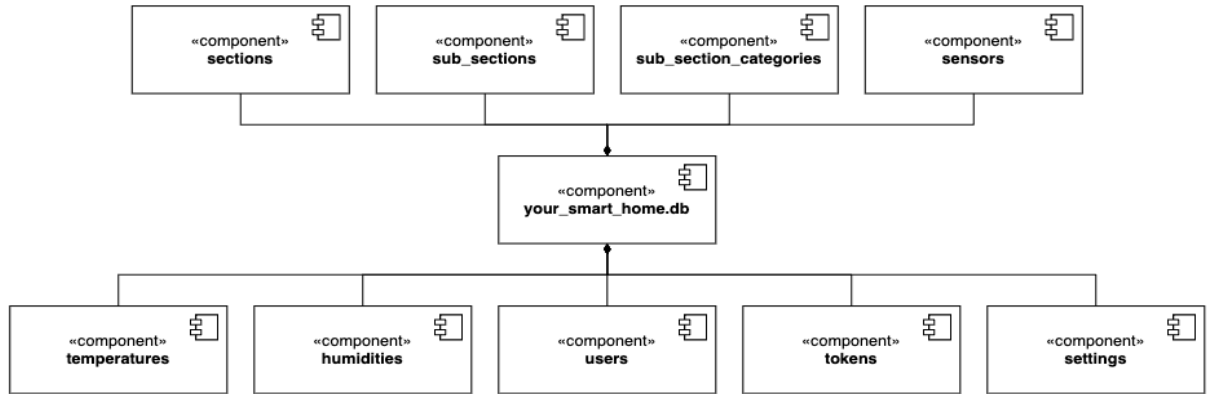
ER diagramma ir datu modeļa veids, kas izmanto attēlojumu, lai parādītu informāciju par datu struktūrām un attiecībām starp tām. Tas ir efektīvs līdzeklis datu bāzu projektēšanā un sapratnes veidošanā par datu organizāciju. ER diagrammas palīdz izstrādāt vienotu un vizuāli viegli saprotamu attēlojumu datu struktūrām, tos izmantojot, lai modelētu sistēmas datu pamatprincipus (Richard Makara, 2022) (*Your Smart Home* ER diagrammu skatīt 3. attēlā).



3. attēls. *Your Smart Home* datubāzes ER diagramma

4.2. DATUBĀZES KOMPONENŠU DIAGRAMMA

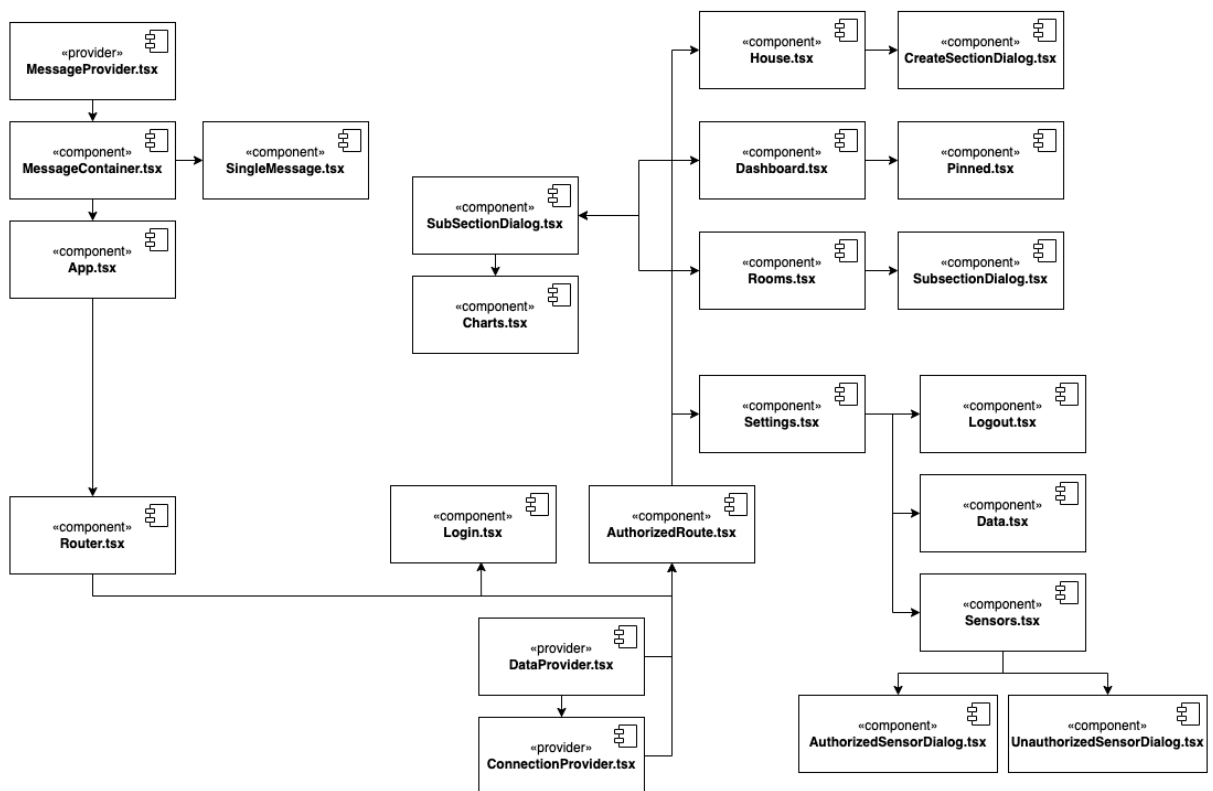
Šī diagramma attēlo pašas datubāzes struktūru, iekļaujot tabulu nosaukumus un savstarpējās relācijas (Visual Paradigm, 2023) (skatīt 4. attēlā).



4. attēls. *Your Smart Home* datubāzes komponentu diagramma

4.3. PRIEKŠGALSISTĒMAS KOMPONENŠU DIAGRAMMA

Šī diagramma attēlo priekšgalsistēmas komponentu savstarpējo mijiedarbību un atkarību (Silpa Sasidharan, 2024) (skatīt 5. attēlā).



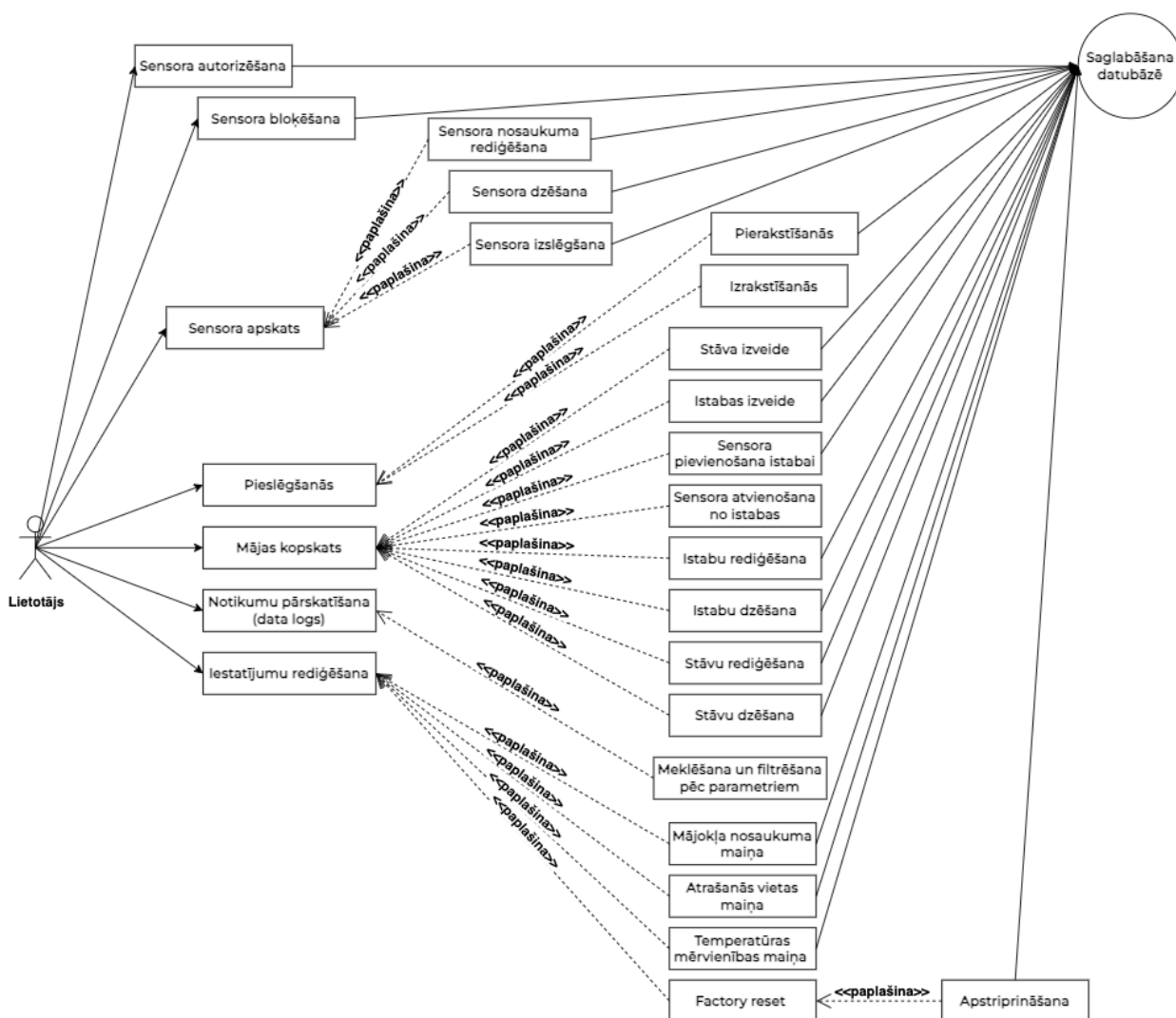
5. attēls. *Your Smart Home* priekšgalsistēmas komponentu diagramma

5. SISTĒMAS DINAMISKAIS UN FUNKCIONĀLAIS MODELIS

Sistēmas dinamiskais un funkcionālais modelis ir datu modeļa veids, kas integrē sistēmas darbības procesus un funkcijas ar laika dimensiju, norādot to mijiedarbību, datu plūsmu, stāvokļu pārmaiņas un lietotāju iesaisti, tādējādi sniedzot holistisku izpratni par sistēmas darbību un funkcionalitāti gan statiskā, gan dinamiskā kontekstā. Šis modelis ir būtisks, lai pilnībā saprastu un efektīvi izstrādātu sistēmu, ņemot vērā gan tās struktūru, gan darbību laikā.

5.1. LIETOŠANAS GADĪJUMU DIAGRAMMA

Lietošanas gadījumu diagramma ir UML (*Unified Modeling Language*) modeļa sastāvdaļa, kas attēlo sistēmas funkcijas un to mijiedarbību ar lietotājiem, identificējot dažādus scenārijus vai lietošanas gadījumus, kā arī tos aktorus, kas iesaistīti katrā gadījumā. Šī diagramma palīdz izprast un dokumentēt sistēmas darbības prasības un funkcionalitāti. *Your Smart Home* lietošanas gadījumu diagramma (skatīt 6. attēlā).



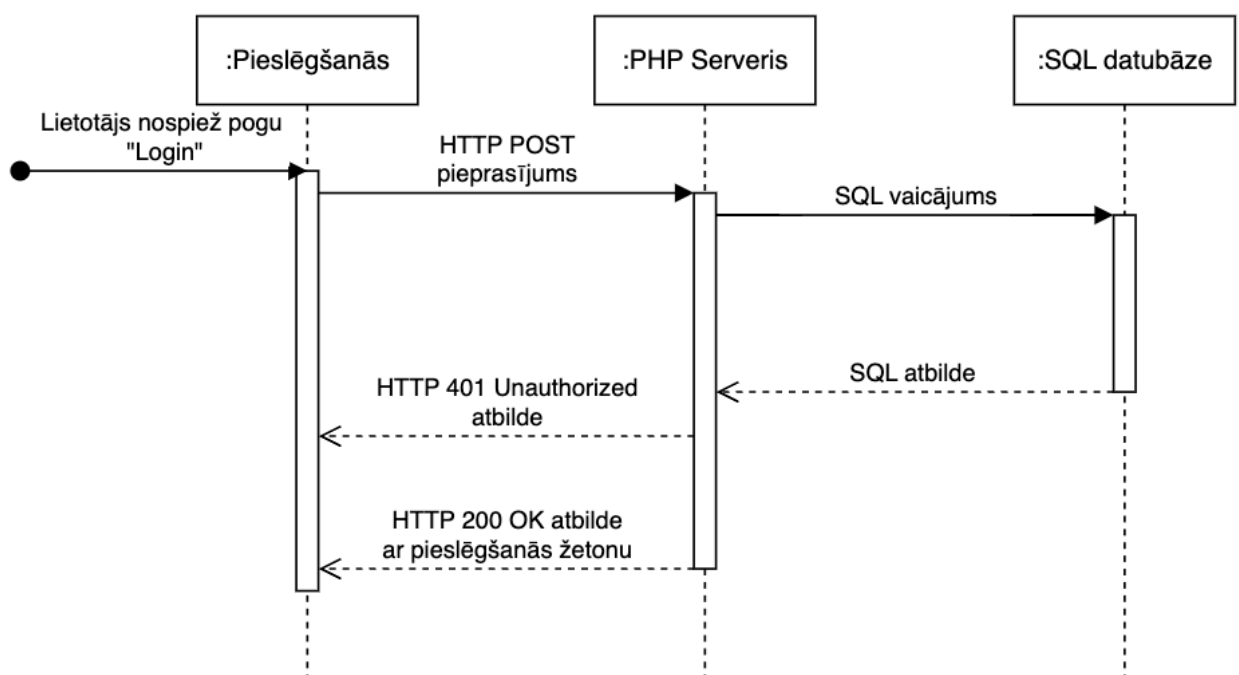
6. attēls. Lietošanas gadījumu diagramma

5.2. SEKVENCES DIAGRAMMA

Sekvences diagramma ir UML (*Unified Modeling Language*) diagramma, kas attēlo mijiedarbību starp objektiem vai sistēmas komponentēm laika secībā, norādot, kā tie mijiedarbojas viens ar otru, izsakot procesus, funkcijas un ziņojumus, kas tiek apmainīti starp šiem objektiem vai komponentēm. Šī diagramma ir noderīga, lai vizuāli izprastu sistēmas darbības secību un komunikāciju starp tās dažādajām sastāvdaļām.

5.2.1. PIESLĒGŠANĀS TĪMEKĻA VIETNEI

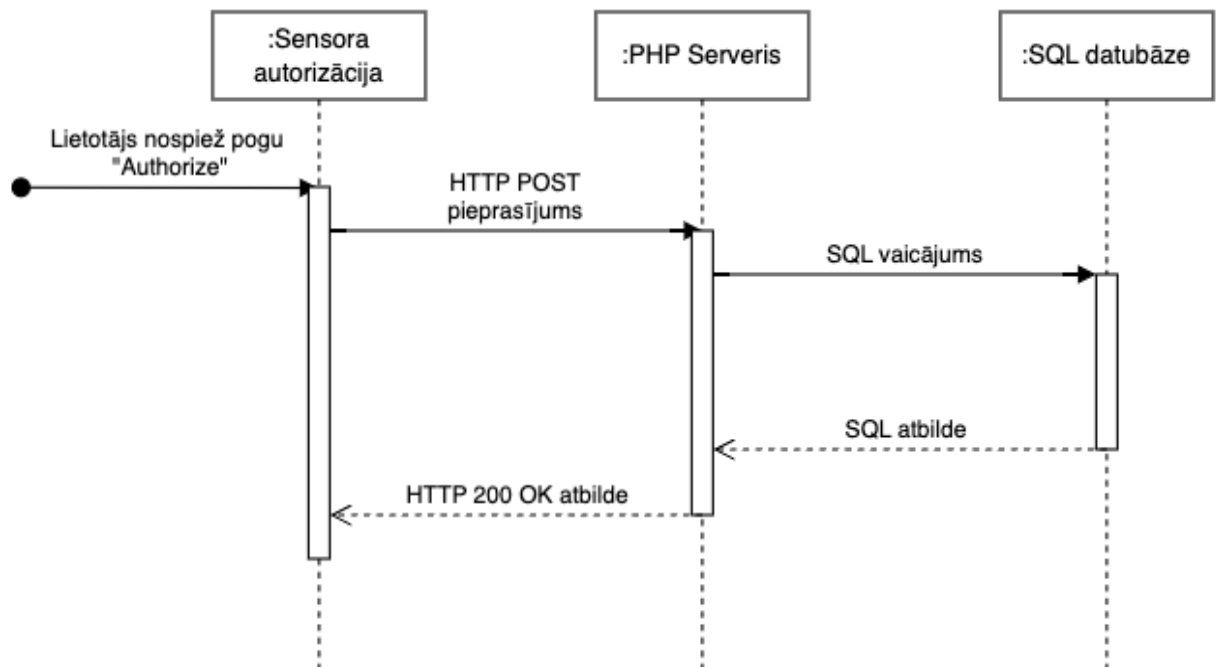
Šajā sekvences diagrammā parādīts process soli pa solim, kad lietotājs *Your Smart Home* tīmekļa lietotnē mēģina autorizēties (skatīt 7. attēlā).



7. attēls. Pieslēgšanās sekvences diagramma

5.2.2. SENSORA AUTORIZĒŠANA TĪMEKĻA VIETNĒ

Šajā sekvences diagrammā tiek parādīts process soli pa solim, kad lietotājs autorizē kādu sensoru, kas pieprasa autorizāciju (skatīt 8. attēlā).



8. attēls. Sensora autorizācijas sekvences diagramma

6. TESTĒŠANA

Testēšana ir būtiska programmatūras izstrādes procesa daļa, nodrošinot, ka risinājums darbojas paredzētajā veidā un atbilst visām prasībām. Testēšanas procesā tiek identificētas un labotas kļūdas, tādējādi uzlabojot programmatūras kvalitāti un lietotāja pieredzi. Šajā nodaļā tiks aprakstīti automātiski un manuālie testi, kas tika izmantoti, lai pārliecinātos par sistēmas funkcionalitāti, veiktspēju un drošību (Peter Koffer, 2022).

6.1. AUTOMĀTISKĀ TESTĒŠANA

Automātiskā testēšana ir būtiska, lai nodrošinātu tīmekļa lietotnes stabilitāti un funkcionalitāti. Izmantojot automātiskos testus, var ātri un efektīvi pārbaudīt vai programma veic savu funkciju atbilstoši prasībām pēc kādām koda izmaiņām. Tas ļoti atvieglo koda uzturēšanu turpmāk, kad tiks veiktas kādas izmaiņas funkcijās, ko pārbauda sarakstītie automātiskie testi. *Your Smart Home* aizmugursistēmas automātiskā testēšana būs balstīta uz *Laravel* ietvara iebūvēto *Feature Tests*, ar kuru tiks pārbaudītas programmas funkcionalitāte un drošība (Smartbear, 2022).

6.1.1. AUTORIZĀCIJAS TESTI

Šajā nodaļā tiks pārbaudītas tīmekļa lietotnes autorizācijas funkcijas gan ar pareizi noformētiem datiem, gan ar nepareizi noformētiem datiem, lai varētu novērtēt kļūdu novēršanas sistēmu. (Autorizācijas automātiskie testu rezultāti aplūkojami 9. attēls. Autorizācijas automātisko testu rezultāti.

Funkcijas kas tiek testētas

- Vai tīmekļa vietnes lietotājs spēj autorizēties ar pareiziem piekļuves datiem.
- Vai tīmekļa vietnes lietotājs nespēj autorizēties ar nepareiziem piekļuves datiem.
- Vai tīmekļa vietnes lietotājs spēj iziet no sava konta.
- Vai tīmekļa vietnes lietotāja autorizācijas žetons tiek pieņemts un atjaunots, ja tas ir pareizi noformatēts. vai nav vecāks par 24 stundām.
- Vai tīmekļa vietnes lietotāja autorizācijas žetons netiek pieņemts un atjaunots, ja tas nav pareizi noformatēts, vai ir vecāks par 24 stundām.

Automātisko testu rezultāti

```
PASS Tests\Feature\AuthApiTest
✓ login user
✓ catch incorrect login details
✓ logout user
✓ token verify ping with correct token
✓ token verification fail with incorrect token
```

6.1.2. SENSORU DARBĪBU TESTI

Šajā nodaļā tiks pārbaudītas sensoru funkcijas un lietotāju darbības ar sensoriem gan ar pareizi noformētiem datiem, gan ar nepareizi noformētiem datiem, lai varētu novērtēt kļūdu novēršanas sistēmu. (Sensoru darbību automātiskie testu rezultāti skatāmi 10. attēlā)

Funkcijas kas tiek testētas

- Vai lietotājs spēj veikt pieprasījumu un saņemt masīvu ar sensoriem, ja ir pareizs piekļuves žetons.
- Vai lietotājs nespēj veikt pieprasījumu un saņemt masīvu ar sensoriem, ja ir nepareizs piekļuves žetons, vai tas ir vecāks par 24 stundām.
- Vai lietotājs spēj veikt pieprasījumu un saņemt masīvu ar neautorizētiem sensoriem, ja ir pareizs piekļuves žetons.
- Vai lietotājs nespēj veikt pieprasījumu un saņemt masīvu ar neautorizētiem sensoriem, ja ir nepareizs piekļuves žetons, vai tas ir vecāks par 24 stundām.
- Vai lietotājs spēj veikt pieprasījumu un saņemt kādu konkrētu sensoru pēc tā savienotās apakšsekcijas identifikācijas numura (id), ja ir pareizs piekļuves žetons.
- Vai lietotājs nespēj veikt pieprasījumu un saņemt kādu konkrētu sensoru pēc tā savienotās apakšsekcijas identifikācijas numura (id), ja ir nepareizs piekļuves žetons, vai tas ir vecāks par 24 stundām.
- Vai sensors spēj veikt autorizācijas pieprasījumu serverim.
- Vai lietotājs ar pareizu piekļuves žetonu spēj autorizēt sensoru, kas pieprasa autorizāciju serverim.
- Vai lietotājs ar nepareizu piekļuves žetonu nespēj autorizēt sensoru, kas pieprasa autorizāciju serverim.
- Vai lietotājs ar pareizu piekļuves žetonu spēj rediģēt sensora nosaukumu.
- Vai lietotājs ar nepareizu piekļuves žetonu nespēj rediģēt sensora nosaukumu.
- Vai lietotājs ar pareizu piekļuves žetonu spēj noņemt sensoru.
- Vai lietotājs ar nepareizu piekļuves žetonu nespēj noņemt sensoru.
- Vai autorizēts sensors spēj sūtīt datus uz serveri.
- Vai neautorizēts serveris nespēj sūtīt datus uz serveri.

Automātisko testu rezultāti

```
PASS Tests\Feature\SensorApiTest
✓ get sensors with valid token
✓ deny access to get sensors with invalid token
✓ get unassigned sensors with valid token
✓ deny access to get unassigned sensors with invalid token
✓ get sensor by subsection with valid token
✓ deny access to get sensor by subsection with invalid token
✓ attempt handshake
✓ accept handshake with valid token
✓ deny access to accept handshake with invalid token
✓ send sensor data with authorized sensor
✓ deny access to send sensor data with unauthorized sensor
✓ edit sensor with valid token
✓ deny access to edit sensor with invalid token
✓ delete sensor with valid token
✓ deny access to delete sensor with invalid token
```

10. attēls. Sensoru automātisko testu rezultāti.

6.1.3. VIEDMĀJOKĻA SEKCIJU TESTI

Šajā nodaļā tiks pārbaudītas viedmājokļa sekciju (stāvu) funkcijas un lietotāju darbības ar tām. Līdzīgi kā sensoru testiem, arī šiem testiem tiks pārbaudīta žetonu autorizācija, jo neviena no šīm funkcijām nav jābūt pieejamai neautorizētam lietotājam. Viedmājokļu sekciju testu rezultāti skatāmi 11. attēlā.

Funkcijas kas tiek testētas

- Vai lietotājs ar pareizu piekļuves žetonu spēj veikt pieprasījumu un saņemt masīvu ar eksistējošajām sekcijām.
- Vai lietotājs ar nepareizu piekļuves žetonu nespēj veikt pieprasījumu un saņemt masīvu ar eksistējošām sekcijām.
- Vai lietotājs ar pareizu piekļuves žetonu spēj veikt pieprasījumu un saņemt konkrētu sekciju meklējot pēc unikālā indentifikatora (id).
- Vai lietotājs ar pareizu piekļuves žetonu, veicot pieprasījumu ar neeksistējošu unikālo identifikatoru (id) saņem atbildē atbilstošo HTTP statusa kodu.
- Vai lietotājs ar pareizu piekļuves žetonu spēj izveidot jaunu sekciju.
- Vai lietotājs ar nepareizu piekļuves žetonu nespēj izveidot jaunu sekciju.
- Vai lietotājs ar pareizu piekļuves žetonu spēj rediģēt esošu sekciju.
- Vai lietotājs ar nepareizu piekļuves žetonu nespēj rediģēt esošu sekciju.
- Vai lietotājs ar pareizu piekļuves žetonu spēj izdzēst esošu sekciju.
- Vai lietotājs ar nepareizu piekļuves žetonu nespēj izdzēst esošu sekciju.

Automātisko testu rezultāti

```
PASS Tests\Feature\SectionApiTest
✓ get sections with valid token
✓ deny access to get sections with invalid token
✓ get single section by id with valid token
✓ deny access to get single section by id with invalid token
✓ catch get single section with invalid id error
✓ create section with valid token
✓ edit section with valid token
✓ deny access to edit section with invalid token
✓ delete section with valid token
✓ unable to delete section with invalid section id
```

11. attēls. Sekciju automātisko testu rezultāti

6.2. TĪMEKĻA LIETOTNES PĀRLŪKPROGRAMMAS PIEEJAMĪBAS TESTI

Tīmekļa lietotnes pieejamība pārlūkprogrammās ir viens no būtiskākajiem aspektiem lietotnes funkcionalitātē. Jo vairāk pārlūkprogrammās pieejama tīmekļa lietotne, jo plašākam klāstam lietotāju risinājums ir pieejams. Visbiežākās problēmas rodas ar vizuālajām (CSS) anomālijām uz dažādām pārlūkprogrammām un dažādu dimensiju ierīcēm (Shreya Bose, 2023). Manuālās pārlūkprogrammas testi tiks apkopoti par *Google Chrome*, *Firefox*, *Safari*, *Microsoft Edge*.

Testi

Saraksts ar veiktajiem pārlūkprogrammas pieejamības testiem:

- CSS anomāliju meklēšana starp pārlūkprogrammām,
- tīmekļa vietnes pārbaude starp pārlūkprogrammām uz dažādām reālām ierīcēm un arī attiecīgo pārlūkprogrammu displeja dimensiju emulatoriem iekš *Inspect Element*.

Rezultāti

Testu laikā netika konstatēta neviena problēma. Pateicoties *TailwindCSS* neviena stila anomālija netika konstatēta starp pārlūkprogrammām, kā arī testu laikā ar dažādām ierīcēm virs 320 pikseļiem displeja platuma netika konstatēta neviens skats kas vizuāli salūzt.

7. LIETOTĀJA CEĻVEDIS

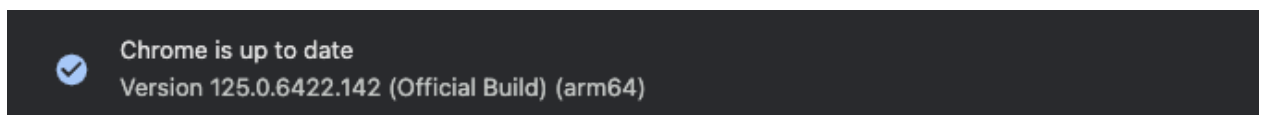
Lietotāja ceļvedis ir būtiska sadaļa, kas sniedz detalizētas instrukcijas un norādījumus par to, kā efektīvi izmantot tīmekļa lietotni. Tas ietver soli pa solim aprakstus un ilustrācijas, kas palīdz lietotājiem izprast sistēmas funkcionalitāti, navigāciju un dažādu rīku izmantošanu. Ceļvedis aptver *Your Smart Home* funkcijas, sākot no pieslēgšanās procesa, līdz pat datu ievadei, sensoru pievienošanai un datu analīzei.

Šis ceļvedis ir svarīgs, lai nodrošinātu, ka lietotāji var ātri un viegli saprast, kā izmantot sistēmu, tādējādi palielinot lietotnes lietošanas efektivitāti un lietotāju apmierinātību. Šajā lietotāja ceļvedī visos vizuālajos piemēros tiek izmantota tīmekļa pārlūkprogramma *Google Chrome* (tīmekļa pārlūkprogrammas versiju skatīt 12. attēlā).

Saite uz tīmekļa lietotni ir <https://home.kantanudarznieciba.lv/>. Saite uz tīmekļa lietotnes pirmkodu iekš *GitHub*: <https://github.com/xHithy/your-smart-home>.

Tīmekļa lietotnes testēšanas profila pieslēgšanās dati:

- lietotājsvārds: TestUser123
- parole: PareizaParole123!



12. attēls. Vizuālajos piemēros izmantotā *Google Chrome* versija

7.1. TĪMEKĻA LIETOTNES ATVĒRŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi atvērtu tīmekļa lietotni. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

Lai atvērtu tīmekļa lietotni, lietotājam nepieciešams:

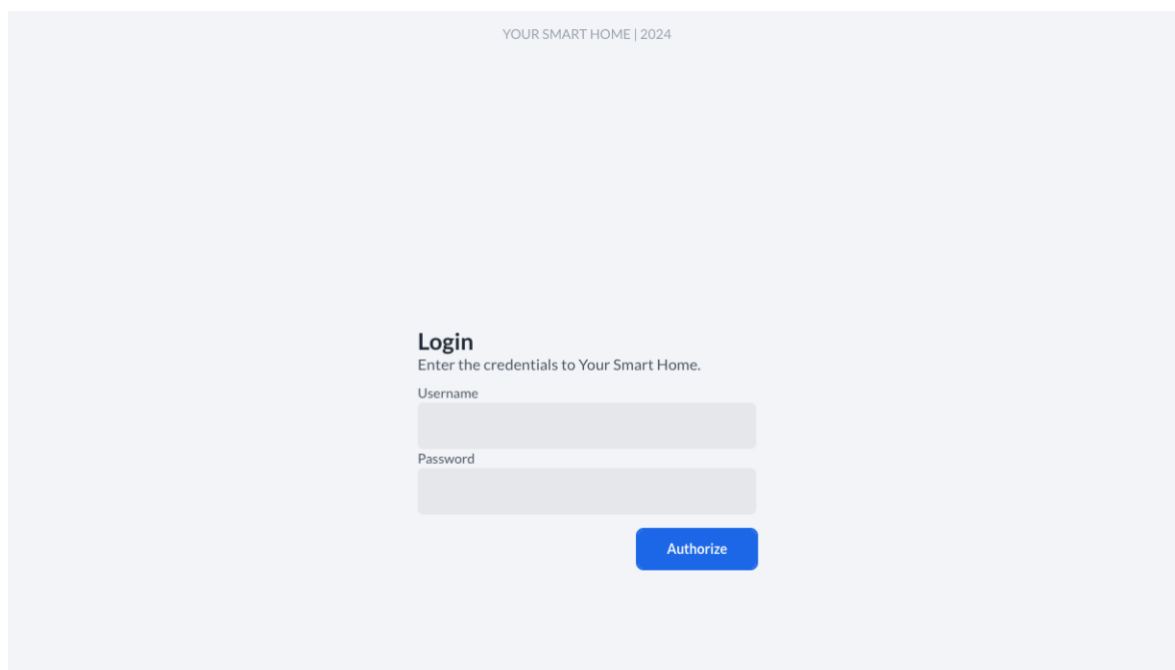
- atvērt kādu datorā pieejamo tīmekļa pārlūkprogrammu (*Google Chrome*, *Safari*, *Firefox*, *Internet Explorer*, *Microsoft Edge*),
- meklēšanas joslā ievadīt tīmekļa lietotnes saiti (skatīt 13. attēlā),



13. attēls. *Google Chrome* pārlūkprogrammas meklēšanas josla ar tīmekļa lietotnes saiti

- uz tastatūras nospiediet pogu “Enter”, lai sāktu tīmekļa lietotnes ielādi.

Kad tīmekļa lietotne būs ielādējusies, ekrānā jāparādās pieslēgšanas skatam (skatīt 14. attēlā).

The screenshot shows a web interface for 'YOUR SMART HOME | 2024'. In the center, there is a 'Login' section with the instruction 'Enter the credentials to Your Smart Home.' Below this, there are two input fields: 'Username' and 'Password'. To the right of the 'Password' field is a blue button labeled 'Authorize'.

14. attēls. *Your Smart Home* pieslēgšanās skats

7.2. PIESLĒGŠANĀS TĪMEKĻA LIETOTNEI

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi pieslēgtos, jeb autorizētos tīmekļa lietotnei. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Kad tīmekļa lietotne ir atvērta, tad pārlūkprogrammas centrā būs ievadlauki un poga *Authorize* (latviešu v. *Autorizēties*). Ir izņēmuma gadījums, ja lietotājs ir jau bijis autorizējies ne vēlāk kā 24 stundas atpakaļ, tādā gadījumā autorizācijas žetons ir saglabāts tīmekļa lietotnes lokālajā kešatmiņā un lietotājs automātiski tiks pāradresēts uz tīmekļa lietotnes sāukmlapu (<http://home.kantanudarznieciba.lv/dashboard>)

Izpildes soļi

Lai pieslēgtos tīmekļa lietotnei lietotājam ir nepieciešams:

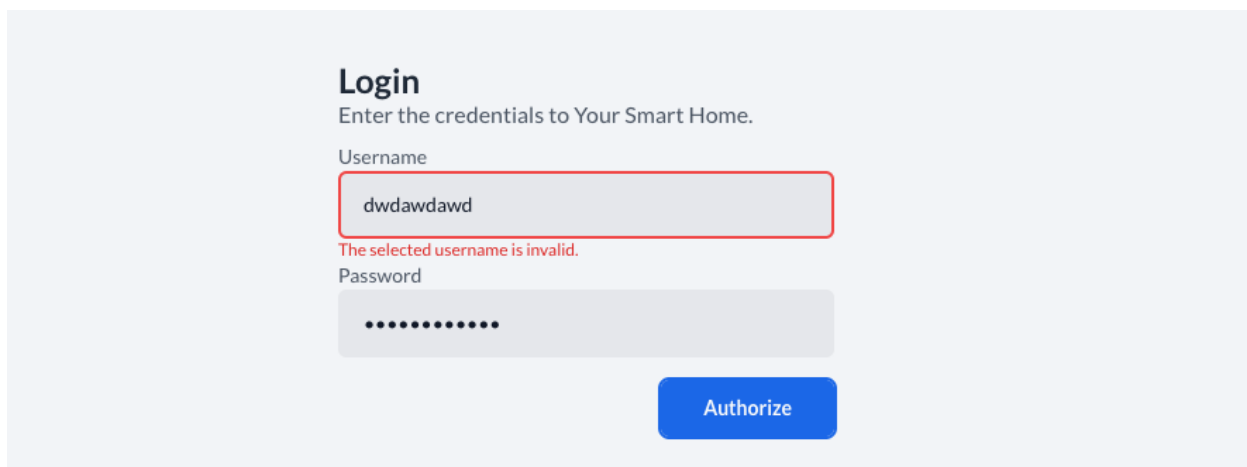
- aizpildīt *Username* (latviešu v. *lietotājevārds*) ievadlauku ar piešķirto lietotājevārdu,
- aizpildīt *Password* (latviešu v. *parole*) ievadlauku ar piešķirto paroli,
- nospiegt pogu *Authorize* (latviešu v. *autorizēties*).

Rezultāts

Ja ievadlauki būs aizpildīti ar pareizajiem piekļuves datiem, tad lietotājs tiks pāradresēts uz lietotnes sāukmlapu (<http://home.kantanudarznieciba.lv/dashboard>). Savukārt, ja kāds no ievadlaukiem neatbilst pareizajiem piekļuves datiem, tad nepareizais ievadlauks būs apvilkt

sarkanā krāsā un zem tā būs paskaidrota ievadlauka kļūda. (skatīt 15. attēlā kļūdas ziņojuma piemēru, ja lietotājvārds neeksistē)

Vizuālā piemēra ekrānšāviņš



15. attēls. Kļūdu ziņojums nepareizi noformatētam lietotājvārda ievadlaukam

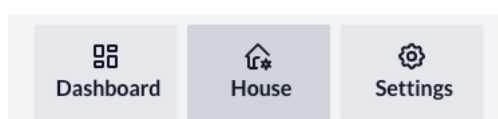
7.3. NAVIGĀCIJAS JOSLAS LIETOŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu navigēt cauri tīmekļa lietotnes skatiem. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Svarīgi zināt, ka navigācijas joslā aktīvais skats, jeb atvērtais skats tiek iezīmēts tumši pelēkā krāsā, kamēr neaktīvie skati ir gaiši pelēkā krāsā (skatīt 16. attēlā). Navigācijas josla datora versijā atrodas lapas augšpusē, savukārt, mobilajā versijā tā atrodama ekrāna apakšā, lai būtu ērtāk lietot vienas rokas režīmā (skatīt 17. attēlā).



16. attēls. Datora navigācijas joslā aktīvs skats *House*



17. attēls. Telefona navigācijas joslā aktīvs skats *House*

Izpildes soļi

- Lai atvērtu skatu *House* (latviešu v. *māja*), lietotājam navigācijas joslā jānoklikšķina poga *House* kurai blakus ir ikona – māja ar zobratu (skatīt 18. attēlā).



18. attēls. Navigācijas joslā neaktīvs skats *House*

- Lai atvērtu skatu *Dashboard* (latviešu v. *Sākuma panelis*), lietotājam navigācijas joslā jānoklikšķina poga *Dashboard* kurai blakus ir ikona – 4 nevienādi taisnstūri (skatīt 19. attēlā).



19. attēls. Navigācijas joslā neaktīvs skats *Dashboard*

- Lai atvērtu skatu *Settings* (latviešu v. *iestatījumi*), lietotājam navigācijas joslā jānoklikšķina poga *Settings* kurai blakus ir ikona – zobrats (skatīt 20. attēlā).



20. attēls. Navigācijas joslā neaktīvs skats *Settings*

7.4. SEKCIJU APSKATE

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu apskatīt *Your Smart Home* sekciju sarakstu. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

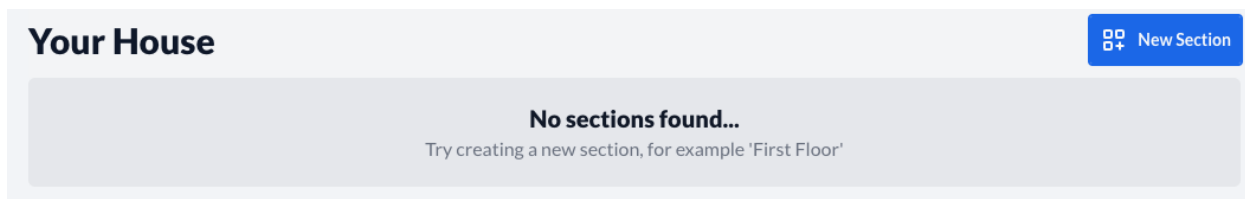
Lai apskatītu sekcijas lietotājam nepieciešams:

- navigācijas joslā noklikšķināt pogu *House* (latviešu v. *māja*).

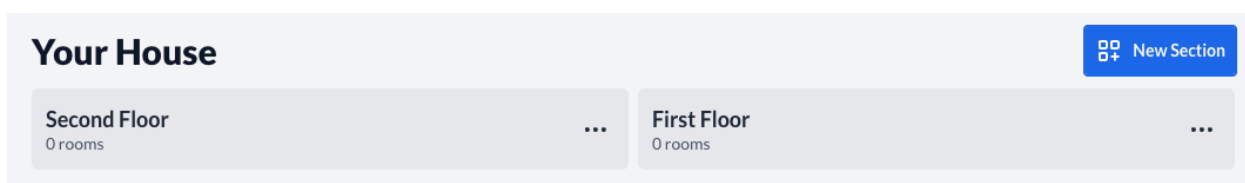
Rezultāts

Kad ielādējas skats *House* (latviešu v. *māja*) un nav neviena sekcija, tad sekciju vietā parādīsies ziņojums, ka nav neviena sekcija atrasta (skatīt 21. attēlā). Savukārt, ja sekcijas tiek atrastas, tās tiks attēlotas taisnstūra laukumos ar nosaukumu, apakšsekciju skaitu un trīspunktu pogu ar kuru var atvērt darbību modālo logu (skatīt 22. attēlā).

Vizuālo piemēru ekrānšāviņi



21. attēls. Ziņojums kad nav atrasta neviena sekcija



22. attēls. *House* skats ar atrastajām sekcijām

7.5. SEKCIJAS PIEVIENOŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu pievienot sekciju *Your Smart Home* sekciju sarakstā. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

Lai pievienotu sekciju, lietotājam nepieciešams:

- atrasties *House* skatā,
- noklikšķināt pogu *New Section* (latviešu v. *jauna sekcija*) (skatīt 23. attēlā),
- pievienošanas modālajā logā aizpildīt *Section name* (latviešu v. *sekcijas nosaukums*) ievadlauku (skatīt 24. attēlā),
- noklikšķināt pogu *Finish* (latviešu v. *pabeigt*) (skatīt 25. attēlā).

Rezultāts

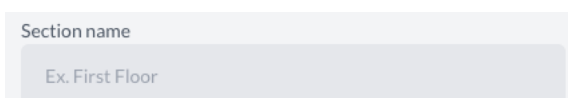
Kad pievienošanas modālajā logā tiek nospiesta poga *Finish* (latviešu v. *pabeigt*), tad modālais logs aizveras, bez lapas pārlādes sekciju sarakstā parādās jaunā sekcija un tīmekļa lietotnes labajā augšējā stūrī parādās zaļš paziņojums, ka sekcija veiksmīgi pievienota (skatīt 26. attēlā). Savukārt, ja *Section name* (latviešu v. *sekcijas nosaukums*) ievadlauks ir nepareizi noformēts, tad tas tiek apvilkt sarkans un zem tā parādās sarkans teksts, kas paskaidro kļūdu (skatīt 27. attēlā).

Vizuālo piemēru ekrānšāviņi

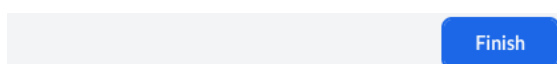
Your House

New Section

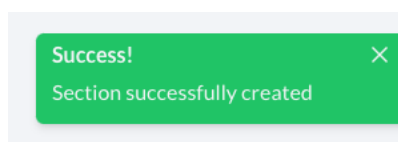
23. attēls. Jaunas sekcijas pievienošanas poga



24. attēls. Pievienošanas modālā loga ievadlauks



25. attēls. Pievienošanas modālā loga poga



26. attēls. Veiksmīgas darbības modālais logs

27. attēls. Sekcijas pievienošanas modālā loga ievadlauka kļūda

7.6. SEKCIJAS DZĒŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu izdzēst kādu sekciju no *Your Smart Home* sekciju saraksta. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

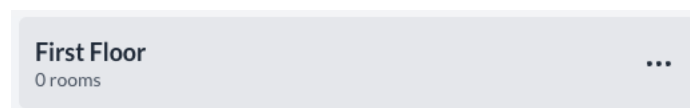
Lai izdzēstu kādu sekciju, lietotājam nepieciešams:

- atrasties *House* skatā,
- sekciju sarakstā atrast sekciju, kuru vēlas izdzēst,
- nospiegt trīspunktu ikonu sekcijas taisnstūra labajā malā (skatīt 28. attēlā),
- uznirstošajā iespēju logā noklikšķināt pogu *Delete* (latviešu v. *dzēst*) (skatīt 29. attēlā),
- uznirstošajā modālajā logā, kur tiek prasīts apstiprinājums, vai lietotājs tiešām vēlas dzēst sekciju, noklikšķināt pogu *Delete* (skatīt 30. attēlā)

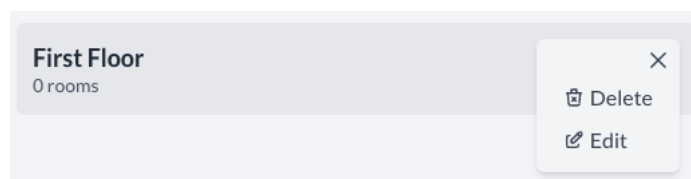
Rezultāts

Kad apstiprinošajā modālajā logā tiek nospiesta poga *Delete*, tad modālais logs aizveras, no sekciju saraksta automātiski pazūd izdzēstā sekcija bez lapas pārlādes un tīmekļa lietotnes labajā augšējā stūrī parādas zaļš modālais logs ar paziņojumu ka sekcija veiksmīgi izdzēsta (skatīt 31. attēlā).

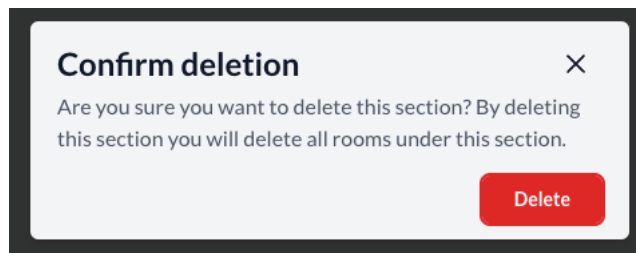
Vizuālo piemēru ekrānšāviņi



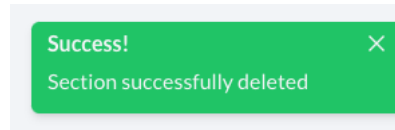
28. attēls. Sekcijas taisnstūris ar trīspunktu ikonu labajā malā



29. attēls. Izvēles logs ar sekcijas dzēšanas opciju



30. attēls. Modālais dzēšanas apstiprinājuma logs



31. attēls. Veiksmīgas sekcijas izdzēšanas paziņojuma modālais logs

7.7. SEKCIJAS REDIGĒŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu rediģēt kādu sekciju no *Your Smart Home* sekciju saraksta. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

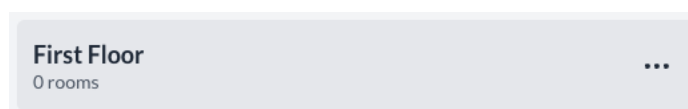
Lai rediģētu kādu sekciju, lietotājam nepieciešams:

- atrasties *House* skatā,
- sekciju sarakstā atrast sekciju kuru vēlas rediģēt,
- nospiegt trīspunktu ikonu sekcijas taisnstūra labajā malā (skatīt 32. attēlā),
- uznirstošajā iespēju logā noklikšķināt pogu *Edit* (latviešu v. *rediģēt*) (skatīt 33. attēlā),
- uznirstošajā rediģēšanas modālajā logā rediģēt sekcijas nosaukumu *Section name* (latviešu v. *sekcijas nosaukums*) ievadlauku (skatīt 34. attēlā),
- uznirstošajā rediģēšanas modālajā logā noklikšķināt pogu *Finish* (latviešu v. *pabeigt*).

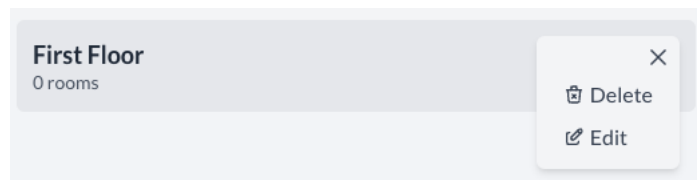
Rezultāts

Ja rediģētais sekcijas nosaukums atbilst noformēšanas prasībām, tad rediģēšanas modālais logs aizvērsies, sekciju sarakstā bez lapas pārlādes būs redzams rediģētais sekcijas nosaukums un tīmekļa lietotnes labajā augšējā malā parādīsies zaļš modālais logs ar paziņojumu ka sekcija veiksmīgi rediģēta (skatīt 35. attēlā). Savukārt, ja rediģētajā sekcijas nosaukumā tiek konstatēta noformēšanas kļūda, tad ievadlauks tiks apvilks sarkanā krāsā un zem tā būs sarkans paskaidrojošais teksts ar kļūdu (skatīt 36. attēlā).

Vizuālo piemēru ekrānšāviņi



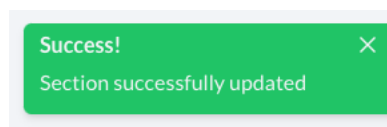
32. attēls. Sekcijas taisnstūris ar trīspunktu ikonu labajā malā



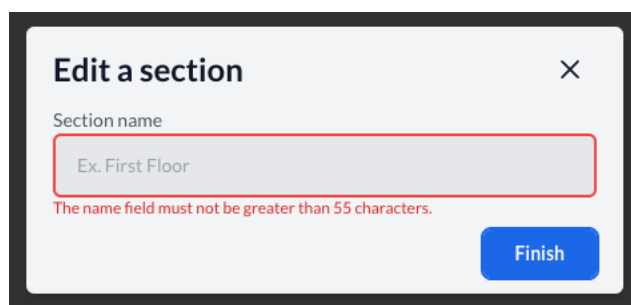
33. attēls. Izvēles logs ar rediģēšanas opciju



34. attēls. Rediģēšanas modālais logs ar ievadlauku un pogu



35. attēls. Veiksmīgas sekcijas rediģēšanas paziņojuma modālais logs



36. attēls. Sekcijas rediģēšanas modālā loga ievadlauka kļūda

7.8. APAKŠSEKCIJU APSKATE

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu apskatīt *Your Smart Home* apakšsekciju sarakstu. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Svarīgi zināt, ka pilnu apakšsekciju sarakstu nav iespējams iegūt, jo apakšsekcijas ir saistītas ar sekcijām, tādēļ iespējams tikai iegūt sarakstu ar apakšsekcijām kas pieder kādai konkrētai sekcijai (apskatīt viena stāva visas telpas).

Izpildes soļi

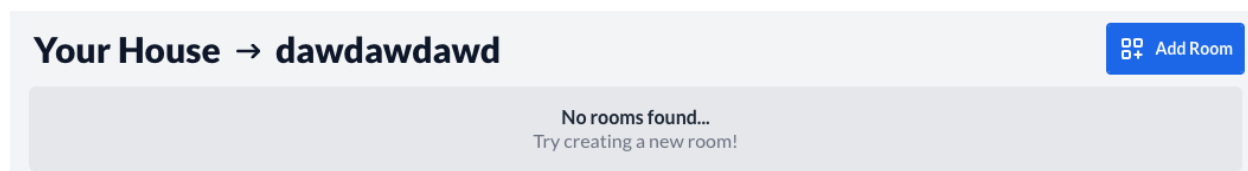
Lai apskatītu kādas sekcijas apakšsekcijas, lietotājam nepieciešams:

- atrasties *House* skatā,
- noklikšķināt uz sekcijas kuras apakšsekcijas vēlas apskatīt.

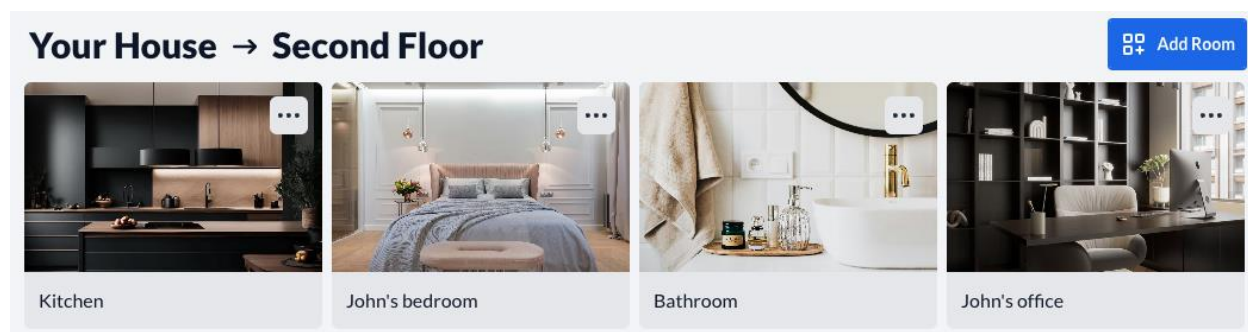
Rezultāts

Kad ielādējas skata *House* (latviešu v. *māja*) apakšskats un nav atrasta neviena apakšsekcija, tad apakšsekciju vietā parādīsies ziņojums, ka nav neviena apakšsekcija atrasta (skatīt 37. attēlā). Savukārt, ja apakšsekcijas tiek atrastas, tās tiks attēlotas taisnstūra laukumos ar nosaukumu, apakšsekcijas kategorijas attēlā un trīspunktu pogu ar kuru var atvērt darbību modālo logu (skatīt 38. attēlā).

Vizuālo piemēru ekrānšāviņi



37. attēls. Sekcijas apakšsekciju saraksts kad netiek atrastas apakšsekcijas



38. attēls. Sekcijas apakšsekciju saraksts

7.9. APAKŠSEKCIJAS PIEVIENOŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu pievienot apakšsekciju *Your Smart Home* apakšsekciju sarakstā. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

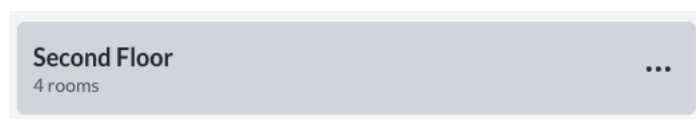
Lai varētu pievienot apakšsekciju, lietotājam nepieciešams:

- noklikšķināt kādu sekciju *House* (latviešu v. *mājas*) skatā (skatīt 39. attēlā),
- noklikšķināt pogu *Add Room* (latviešu v. *pievienot istabu*) (skatīt 40. attēlā),
- apakšsekcijas pievienošanas modālī aizpildīt *Room name* (latviešu v. *istabas nosaukums*) ievadlauku un izvēlēties *Category* (latviešu v. *kategorija*) izvēles lauku (skatīt 41. attēlā),
- noklikšķināt pogu *Finish* (latviešu v. *pabeigt*) (skatīt 42. attēlā).

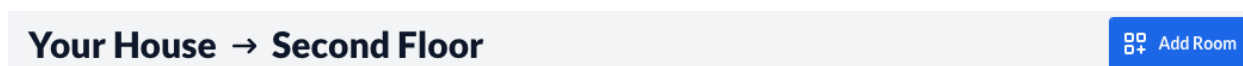
Rezultāts

Ja apakšsekcijas pievienošanas modāli ievadlauks ir pareizi noformēts un izvēles lauks ir izvēlēts, tad apakšsekcijas pievienošanas modāli automātiski aizveras, apakšsekciju sarakstā bez lapas pārlādes parādās jaunā apakšsekcija un tīmekļa lietotnes labajā augšējā malā parādās zaļš modālais logs ar paziņojumu, ka apakšsekcija veiksmīgi pievienota (skatīt 43. attēlā). Savukārt, ja ievadlauks ir nepareizi noformēts, vai arī izvēles logā nav izvēlēta kāda opcija, tad katrs lauks kas nepareizi noformatēts tiek apvilīts sarkanā krāsā un zem tā parādās sarkans teksts ar kļūdas skaidrojumu (skatīt 44. attēlā).

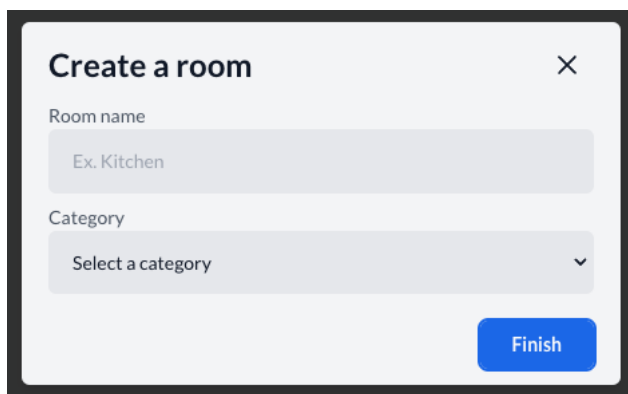
Vizuālo piemēru ekrānšāviņi



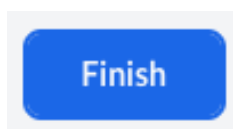
39. attēls. Viena sekcija sekciju sarakstā



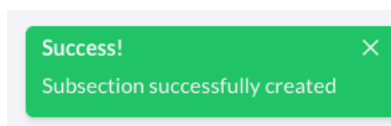
40. attēls. Apakšsekcijas pievienošanas poga labajā malā



41. attēls. Apakšsekcijas pievienošanas modālis



42. attēls. Apakšsekcijas pievienošanas modālā loga pabeigšanas poga



43. attēls. Veiksmīgas apakšsekcijas pievienošanas modālais paziņojums

44. attēls. Apakšsekcijas pievienošanas modālā loga ievadlauku kļūdu ziņojumi

7.10. APAKŠSEKCIJAS REDIĢĒŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu rediģēt konkrētu apakšsekciju *Your Smart Home* apakšsekciju sarakstā. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

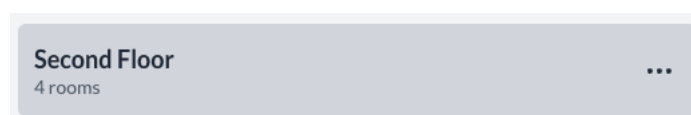
Lai varētu rediģēt konkrētu apakšsekciju, lietotājam nepieciešams:

- noklikšķināt kādu sekciju *House* (latviešu v. *mājas*) skatā (skatīt 45. attēlā),
- noklikšķināt trīspunktu pogu izvēlētajā apakšsekcijā (skatīt 46. attēlā),
- iespēju modāli noklikšķināt pogu *Edit* (latviešu v. *rediģēt*) (skatīt 47. attēlā),
- apakšsekcijas rediģēšanas modālajā logā rediģēt *Room name* (latviešu v. *istabas nosaukums*) ievadlauku vai izvēlēties *Category* (latviešu v. *kategorija*) izvēles lauku (skatīt 48. attēlā),
- apakšsekcijas rediģēšanas modālī noklikšķināt pogu *Finish* (latviešu v. *pabeigt*) (skatīt 48. attēlā).

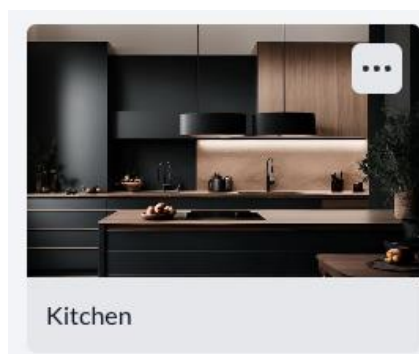
Rezultāts

Ja apakšsekcijas rediģēšanas modālī ievadlauks ir pareizi noformēts un izvēles lauks ir izvēlēts, tad apakšsekcijas pievienošanas modālais logs automātiski aizveras, apakšsekciju sarakstā bez lapas pārlādes parādās rediģētā apakšsekcija un tīmekļa lietotnes labajā augšējā malā parādās zaļš modālais logs ar paziņojumu, ka apakšsekcija veiksmīgi rediģēta (skatīt 49. attēlā). Savukārt, ja ievadlauks ir nepareizi noformēts, vai arī izvēles logā nav izvēlēta neviena opcija, tad katrs lauks kas nepareizi noformatēts tiek apvilīts sarkanā krāsā un zem tā parādās sarkans teksts ar kļūdas skaidrojumu (skatīt 50. attēlā).

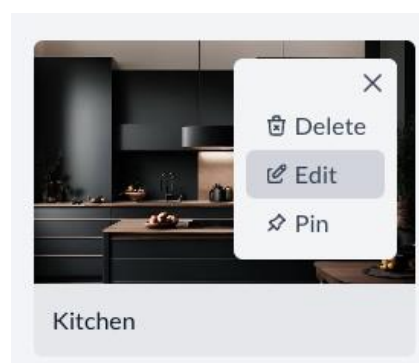
Vizuālo piemēru ekrānšāviņi



45. attēls. Viena sekcija sekciju sarakstā



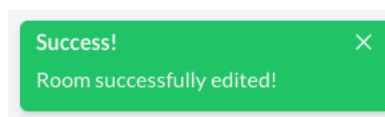
46. attēls. Apakšsekcijas taisnstūra augšējā malā trīspunktu poga



47. attēls. Iespēju modālajā logā iekrāsota rediģēšanas poga

A screenshot of the "Edit room" modal dialog. It has a title bar with "Edit room" and a close button (X). The form contains two input fields: "Room name" with the value "Kitchen" and "Category" with a dropdown menu showing "Kitchen". A blue "Finish" button is located at the bottom right.

48. attēls. Apakšsekciju rediģēšanas modālais logs ar ievadlaukiem un pogu



49. attēls. Veiksmīgas apakšsekcijas rediģēšanas modālais paziņojums

50. attēls. Apakšsekcijas rediģēšanas modālā loga ievadlauku kļūdu ziņojumi

7.11. APAKŠSEKCIJAS DZĒŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu dzēst kādu konkrētu apakšsekciju no *Your Smart Home* apakšsekciju saraksta. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

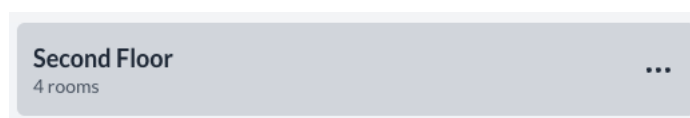
Lai varētu dzēst kādu konkrētu apakšsekciju, lietotājam nepieciešams:

- noklikšķināt kādu sekciju *House* (latviešu v. *mājas*) skatā (skatīt 51. attēlā),
- noklikšķināt trīspunktu pogu izvēlētajā apakšsekcijā (skatīt 52. attēlā),
- iespēju modālī noklikšķināt pogu *Delete* (latviešu v. *dzēst*) (skatīt 53. attēlā),
- apstiprinājuma modālajā logā noklikšķināt pogu *Delete* (latviešu v. *dzēst*) (skatīt 54. attēlā).

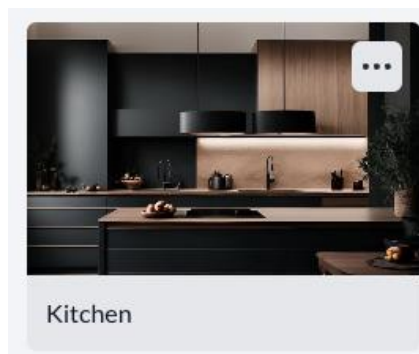
Rezultāts

Kad apstiprinošajā modālajā logā tiek nospiesta poga *Delete* (latviešu v. *dzēst*), tad modālais logs aizveras, no apakšsekciju saraksta bez tīmekļa lietotnes pārlādes pazūd izdzēstā apakšsekcija un tīmekļa lietotnes labajā augšējā stūrī parādās zaļš modālais logs ar paziņojumu ka apakšsekcija veiksmīgi izdzēsta (skatīt 55. attēlā).

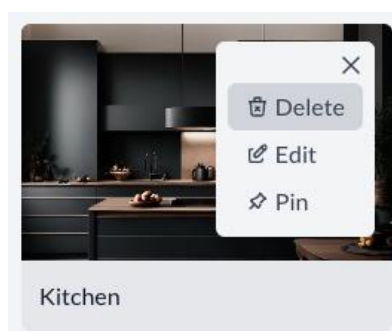
Vizuālo piemēru ekrānšāviņi



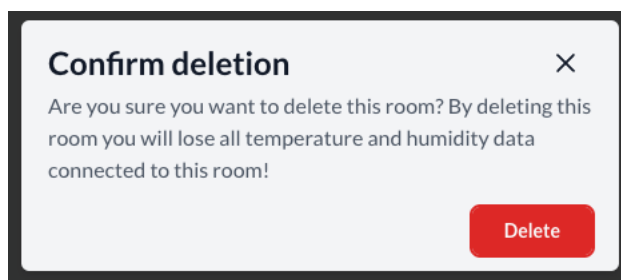
51. attēls. Viena sekcija sekciju sarakstā



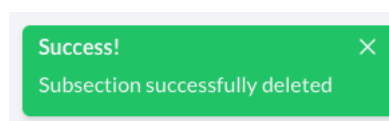
52. attēls. Apakšsekcijas taisnstūra augšējā malā trīspunktu poga



53. attēls. Iespēju modālajā logā iekrāsota dzēšanas poga



54. attēls. Apakšsekcijas dzēšanas apstiprināšanas modālais logs



55. attēls. Apakšsekcijas veiksmīgas izdzēšanas modālais paziņojums

7.12. APAKŠSEKCIJAS PIESPRAUŠANA UN ATSPRAUŠANA NO SĀKUMA PANEĻA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu piespraust vai atspraust kādu konkrētu apakšsekciju *Your Smart Home* sākuma panelī. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi piespraušanai

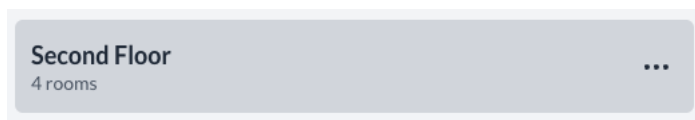
Lai varētu piespraust kādu konkrētu apakšsekciju sākuma panelim, lietotājam nepieciešams:

- noklikšķināt kādu sekciju *House* (latviešu v. *mājas*) skatā (skatīt 56. attēlā),
- noklikšķināt trīspunktu pogu kādā apakšsekcijā kura nav jau piesprausta (skatīt 57. attēlā),
- iespēju modāli noklikšķināt pogu *Pin* (latviešu v. *piespraust*) (skatīt 58. attēlā)

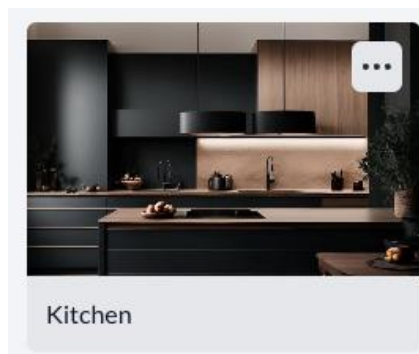
Rezultāts

Kad tiek nospiesta poga *Pin* (latviešu v. *piespraust*), tīmekļa lietotnes labajā augšējā malā parādās zaļš modālais logs ar paziņojumu par veiksmīgu apakšsekcijas piespraušanu (skatīt 59. attēlā), apakšsekcijā kas tika piesprausta parādās piespraudītes ikona (skatīt 60. attēlā), kā arī tagad būs pieejama iespēja šo apakšsekciju atspraust no sākuma paneļa.

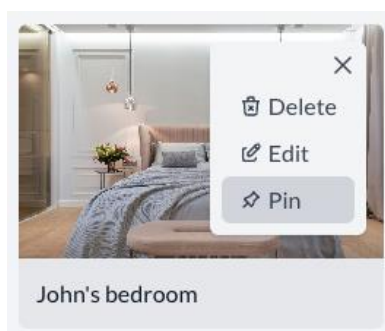
Vizuālo piemēru ekrānšāviņi



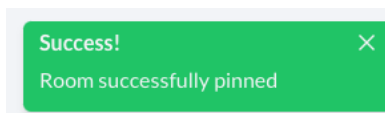
56. attēls. Viena sekcija sekciju sarakstā



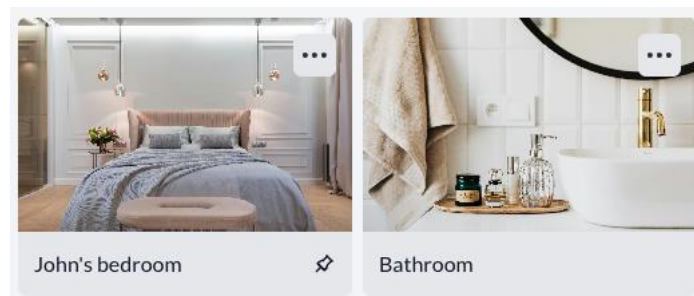
57. attēls. Apakšsekcijas taisnstūra augšējā malā trīspunktu poga



58. attēls. Apakšsekcijas darbību izvēles logā iezīmēta piespraušanas poga



59. attēls. Veiksmīgas apakšsekcijas piespraušanas modālais paziņojums



60. attēls. Piesprausta apakšsekcija kreisajā malā, nepiesprausta labajā

Izpildes soļi atspraušanai

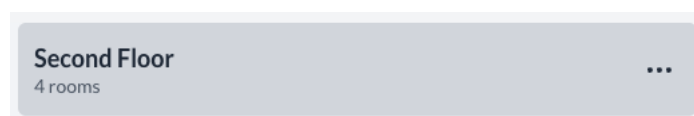
Lai varētu atspraust kādu konkrētu apakšsekciju no sākuma paneļa, lietotājam nepieciešams:

- noklikšķināt kādu sekciju *House* (latviešu v. *mājas*) skatā (skatīt 61. attēlā),
- noklikšķināt trīspunktu pogu kādā apakšsekcijā kurai ir piespraudes ikona (skatīt 62. attēlā),
- iespēju modāli noklikšķināt pogu *Unpin* (latviešu v. *atspraust*) (skatīt 63. attēlā)

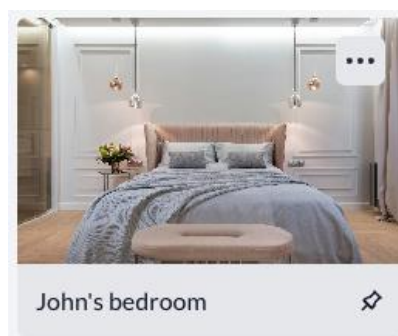
Rezultāts

Kad tiek nospiesta poga *Unpin* (latviešu v. *atspraust*), tīmekļa lietotnes labajā augšējā malā parādās zaļš modālais logs ar paziņojumu par veiksmīgu apakšsekcijas atspraušanu (skatīt 64. attēlā), apakšsekcijā kas tika atsprausta pazūd piespraudītes ikona bez lapas pārlādes (skatīt 65. attēlā), kā arī tagad būs pieejama iespēja šo apakšsekciju piespraust sākuma panelim.

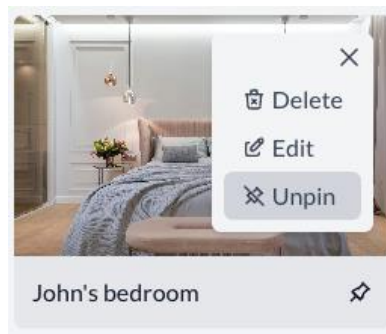
Vizuālo piemēru ekrānšāviņi



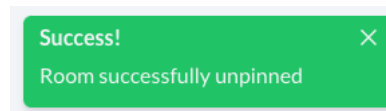
61. attēls. Viena sekcija sekciju sarakstā



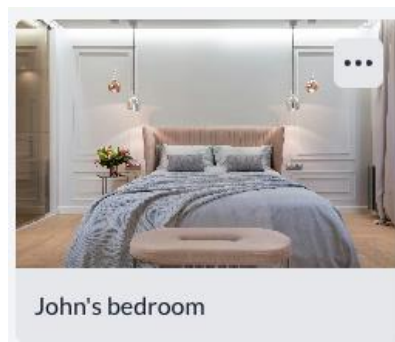
62. attēls. Apakšsekcija ar piespraudes ikonu



63. attēls. Apakšsekcija ar atzīmētu atspaušanas pogu



64. attēls. Veiksmīgas apakšsekcijas atspaušanas modālais paziņojums



65. attēls. Apakšsekcija pēc atspaušanas

7.13. SENSORA PIESAISTĪŠANA APAKŠSEKCIJAI

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi piesaistīt kādu sensoru konkrētajai apakšsekcijai. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

Lai varētu piesaistīt kādu sensoru konkrētai apakšsekcijai, lietotājam nepieciešams:

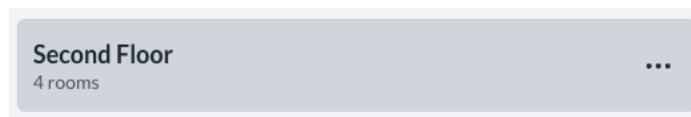
- noklikšķināt kādu sekciju *House* (latviešu v. *mājas*) skatā (skatīt 66. attēlā),
- izvēlēties kādu apakšsekciju un uzklikšķināt uz tās (skatīt 67. attēlā),
- apakšsekcijas datu modālajā logā apakšā atvērt un izvēlēties kādu no autorizētajiem sensoriem kas nav piesaistīti jau kādai apakšsekcijai (skatīt 68. attēlā).

Rezultāti

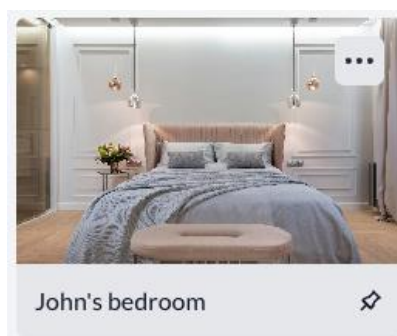
Kad tiek izvēlēts kāds sensors tad tīmekļa lietotnes labajā augšējā malā parādās zaļš modālais logs ar paziņojumu, ka sensors veiksmīgi piesaistīts apakšsekcijai (skatīt 69. attēlā). Ja sensors ir *Online* (latviešu v. *tiešsaistē*), tad ik pēc 30 sekundēm tiks sūtīti dati un tos varēs apskatīt konkrētās apakšsekcijas datu modālajā logā pieejamajos grafikos. Ja sensors ir *Offline* (latviešu v.

bezsaistē), tad netiks sūtīti dati. Savukārt, ja nav bijis pieejams neviens autorizēts sensors, vai arī tie visi ir bijuši jau piesaistīti kādai apakšsekcijai, tad izvēles logs nebūs pieejams (skatīt 70. attēlā).

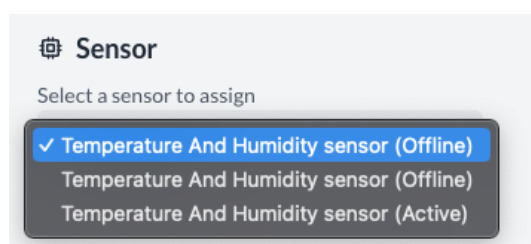
Vizuālo piemēru ekrānšāviņi



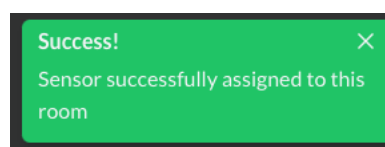
66. attēls. Viena sekcija sekciju sarakstā



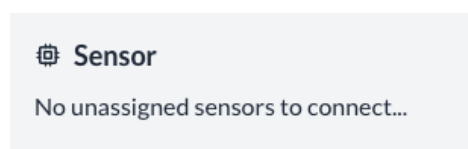
67. attēls. Viena apakšsekcija apakšsekciju sarakstā



68. attēls. Sensoru piesaistīšanas izvēles lauks



69. attēls. Sensora veiksmīga piesaistīšanas modālais paziņojums



70. attēls. Ziņojums, ja nav pieejams neviens autorizēts sensors, kas nav piesaistīts apakšsekcijai

7.14. SENSORA ATSAVINĀŠANA NO APAKŠSEKCIJAS

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi atsavinātu sensoru kādai konkrētajai apakšsekcijai. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

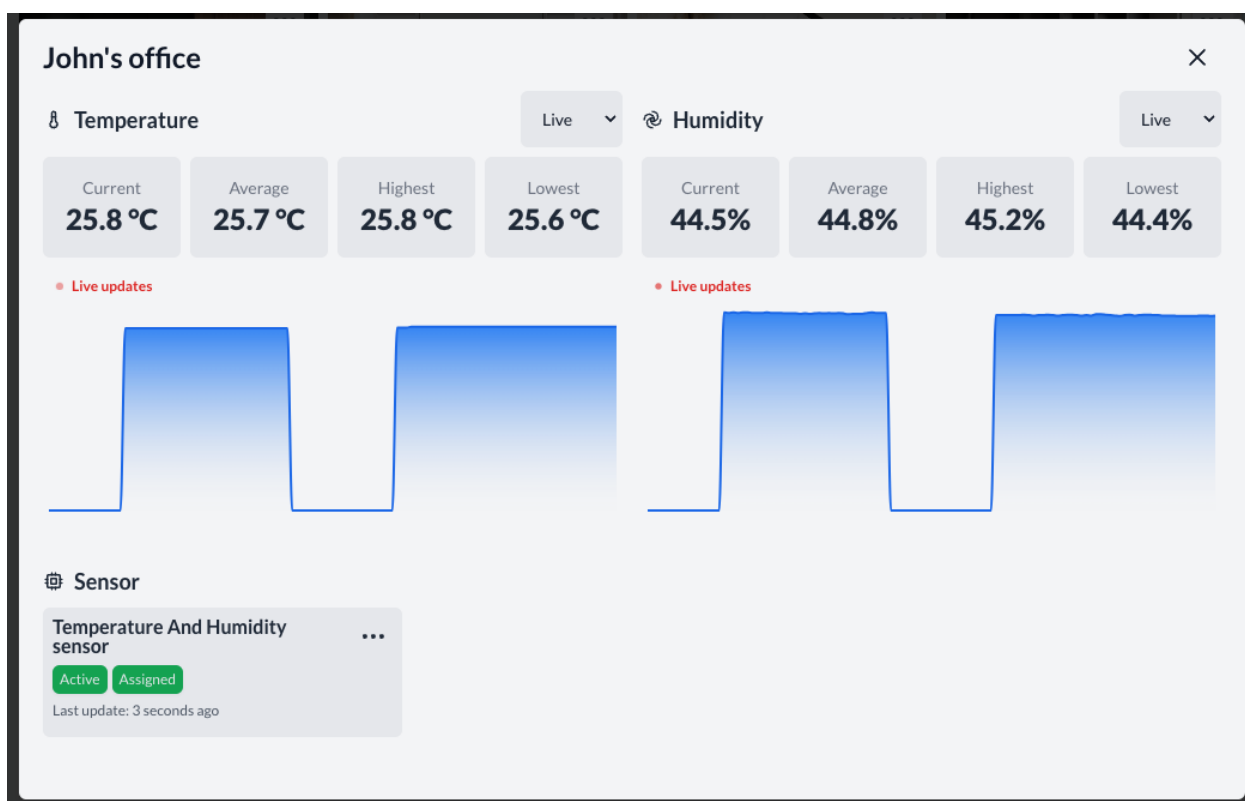
Lai varētu atsavināt sensoru kādai konkrētai apakšsekcijai, lietotājam nepieciešams:

- atrasties kādā apakšsekcijas datu modālajā logā ar piesaistītu sensoru (skatīt 71. attēlā),
- apakšsekcijas datu modālajā logā apakšā pie sensora taisnstūra labajā augšējā stūrī noklikšķināt uz trīspunktu ikonas (skatīt 72. attēlā)
- iespēju modālajā logā noklikšķināt uz pogas *Unassign* (latviešu v. *atsavināt*) (skatīt 73. attēlā)

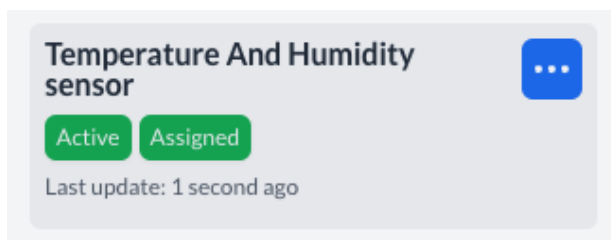
Rezultāti

Kad tiek nospiesta poga *Unassign* (latviešu v. *atsavināt*), tīmekļa lietotnes labajā augšējā stūrī parādās zaļš modālais logs ar paziņojumu, ka sensors veiksmīgi atsavināts no apakšsekcijas (skatīt 74. attēlā).

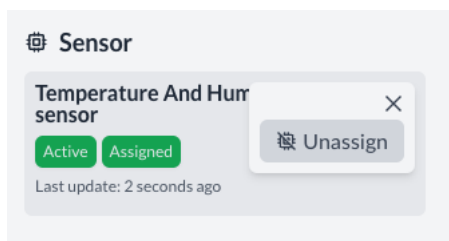
Vizuālo piemēru ekrānšāviņi



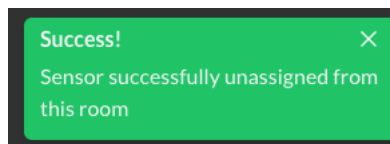
71. attēls. Apakšsekcijas datu modālais logs ar piesaistītu sensoru



72. attēls. Sensora taisnstūris ar iezīmētu trīspunktu pogu



73. attēls. Iespēju modālais logs ar iezīmētu sensora atsavināšanas pogu



74. attēls. Veiksmīgas sensora atsavināšanās modālais paziņojums

7.15. APAKŠSEKCIJU DATU GRAFIKU LAIKA POSMA MAIŅA UN PASKAIDROJUMI

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu mainīt temperatūras un mitruma grafika laika posmus, kā arī tiks paskaidrotas katra laika posma nianse (skatīt 2. tabulu). Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Paskaidrojumi

2. tabula

Datu grafiku laika posmu nianse

Laika posms	Laika posma nianse
<i>Live</i> (latviešu v. <i>reāllaika</i>)	Šis laika posma grafiks satur datus par pēdējo stundu, 30 sekunžu intervālos un ir vienīgais laika posma paveids kas atbalsta reāllaika datu saņemšanu no autorizēta sensora.
<i>Day</i> (latviešu v. <i>diena</i>)	Šis laika posma grafiks satur datus par pēdējām 24 stundām, 10 minūšu intervālos. Šinī grafikā dati netiek atjaunoti reāllaikā.
<i>Week</i> (latviešu v. <i>nedēļa</i>)	Šis laika posma grafiks satur datus par pēdējo nedēļu, vienas stundas intervālos. Šinī grafikā dati netiek atjaunoti reāllaikā.
<i>All Time</i> (latviešu v. <i>visa laika</i>)	Šis laika posma grafiks satur visus vēsturiskos datus no pirmā reģistrētā ieraksta, dienas

	intervālos. Šinī grafikā dati netiek atjaunoti reāllaikā.
--	---

3. tabula

Laika posmu atslēgvārdu skaidrojumi

Atslēgvārds	Paskaidrojums
Intervāls	Grafikā datu intervāls atspoguļojas šādi: dati tiek atjaunoti ik pēc 30 sekundēm noklusēti. Stundas intervāla grafiks no 120 ierakstiem ($30 \text{ sekundes} * 120 = 3600 \text{ sekundes} (1 \text{ stunda})$) aprēķinās tā laika posma vidējo datu vērtību un viena iedaļa grafikā būs šo 120 ierakstu vidējā vērtība.

Izpildes soli

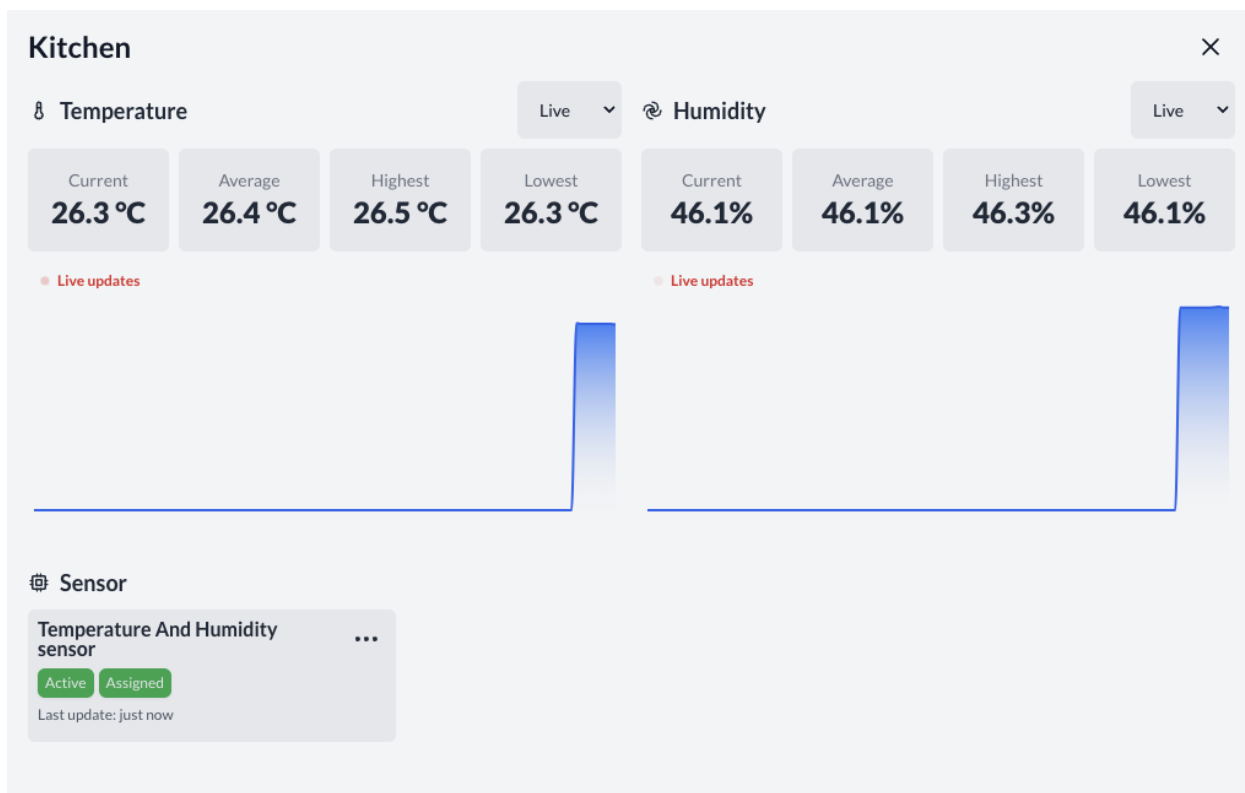
Lai varētu mainīt datu grafika laika posmu, lietotājam nepieciešams:

- atrasties kādā apakšsekcijas datu modālajā logā kurā ir bijis vai ir piesaistīts sensors kurš ir sūtījis vai sūta datus (skatīt 75. attēlā),
- virs mitruma vai temperatūras grafikas noklikšķināt izvēles lauku (skatīt 76. attēlā)
- atvērtajā izvēles laukā izvēlēties grafika laika posmu un noklikšķināt uz to (skatīt 77. attēlā)

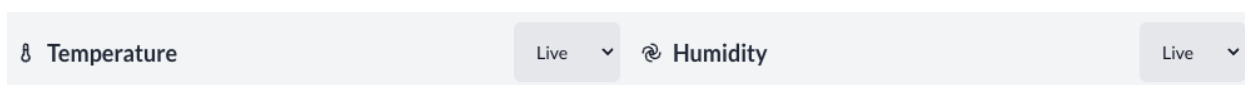
Rezultāts

Ja izvēlētajā laika posmā ir vismaz viens datu ieraksts, tad grafiks ar datiem tiks ielādēts un lietotājs varēs pār to ar kursoru virzīties un skatīties tieši tā intervāla datus (skatīt 78. attēlā). Ja dotajā laika posmā nav bijis neviens ieraksts, tad parādīsies ziņojums, ka nav pieejami dati (skatīt 79. attēlā).

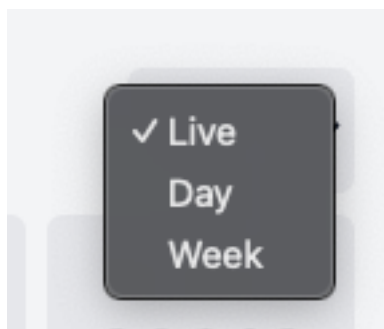
Vizuālo piemēru ekrānšāviņi



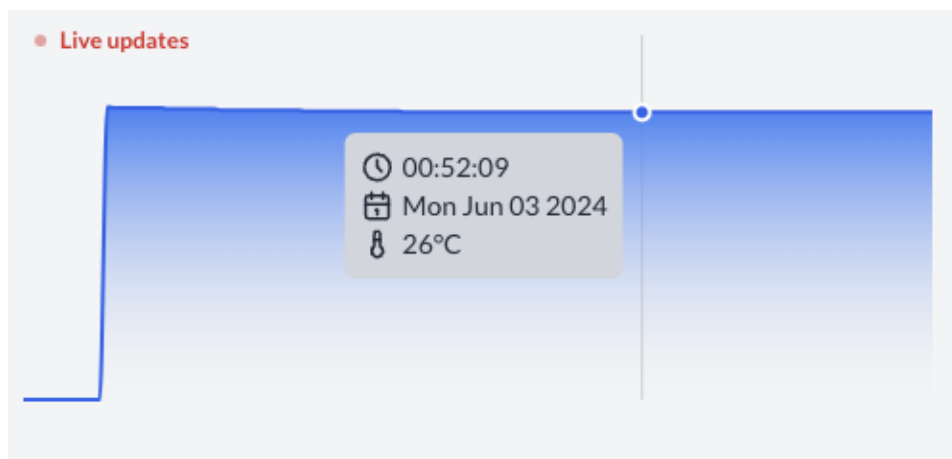
75. attēls. Apakšsekcijas datu modālais logs ar piesaistītu sensoru un datiem



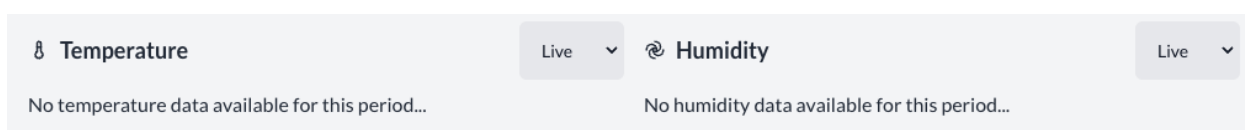
76. attēls. Temperatūras un mitruma grafika laika posma izvēles lauki



77. attēls. Atvērts temperatūras grafika izvēles lauks



78. attēls. Temperatūras grafiks ar datiem kad kursora ir novietots uz grafika iedaļas



79. attēls. Temperatūras un mitruma grafiku vietā paziņojums, ka nav atrasti dati

7.16. NEAUTORIZĒTU SENSORU APSKATE UN AUTORIZĒŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu apskatīt un autorizēt sensorus, kā arī tiks paskaidrotas svarīgas nianšes neautorizēto sensoru sarakstā. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Svarīgi zināt – kad jauns sensors mēģina savienoties ar *Your Smart Home*, tīmekļa lietotnes labajā augšējā malā parādīsies zils modālais paziņojums, ka sensors mēģina savienoties. Modālajā logā būs uzrakstīts sensora nosaukums un ar maziem burtiem sensora unikālais žetons (skatīt 80. attēlā). Sensors kamēr nav autorizēts, sūta autorizācijas pieprasījumu ik pēc 5 sekundēm, to var arī redzēt vizuāli tīmekļa lietotnes saskarnē pie neautorizētajiem sensoriem, zem sensora nosaukuma ir ar maziem pelēkiem burtiem rakstīts kad ir pēdējais autorizācijas pieprasījums veikts (skatīt 81. attēlā), ja pēdējais pieprasījums ir bijis vēlāk par 15 sekundēm, tad sensors pazūd no saraksta līdz tiek veikts jauns pieprasījums.

Izpildes soļi sensoru apskatei

Lai veiksmīgi apskatītu neautorizēto sensoru sarakstu, lietotājam nepieciešams:

- atvērt ar navigācijas joslas palīdzību *Settings* (latviešu v. *iestatījumi*) skatu (skatīt 82. attēlā),
- noklikšķināt pogu *Unauthorized Sensors* (latviešu v. *neautorizēti sensori*) (skatīt 83. attēlā).

Rezultāts

Kad tiek nospiesta poga *Unauthorized Sensors* (latviešu v. *neautorizēti sensori*) atvērsies modālais logs ar neautorizēto sensoru sarakstu. Ja nav neautorizētu sensoru, tad parādīsies

ziņojums (skatīt 84. attēlā), savukārt, ja ir neautorizēti sensori, tos attēlos sarakstā (skatīt 85. attēlā).

Izpildes soļi sensoru autorizēšanai

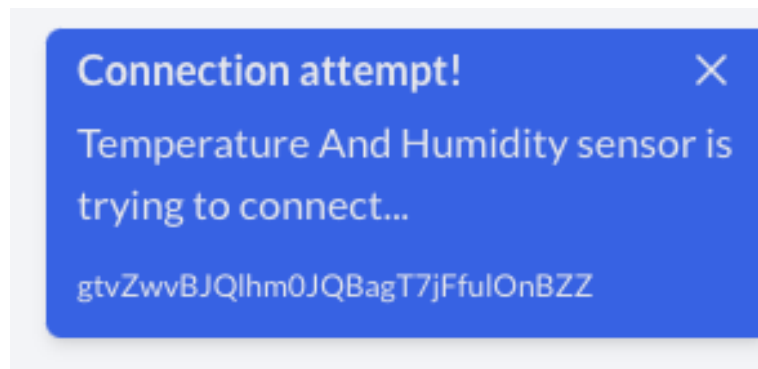
Lai veiksmīgi autorizētu kādu sensoru, lietotājam nepieciešams:

- atvērt ar navigācijas joslas palīdzību *Settings* (latviešu v. *iestatījumi*) skatu (skatīt 82. attēlā),
- noklikšķināt pogu *Unauthorized Sensors* (latviešu v. *neautorizēti sensori*) (skatīt 83. attēlā).
- uz neautorizētā sensora taisnstūra labajā apakšējā malā nospiegt pogu *Authorize* (latviešu v. *autorizēt*) (skatīt 86. attēlā).

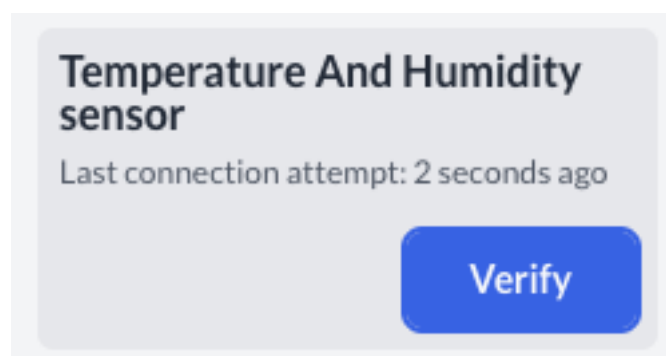
Rezultāts

Kad tiek nospiesta poga *Authorize* (latviešu v. *autorizēt*) modālais logs ar neautorizētajiem sensoriem automātiski aizvērsies, tīmekļa lietotnes labajā augšējā malā parādīsies zaļš modālais logs ar paziņojumu, ka sensors veiksmīgi autorizēts (skatīt 87. attēlā).

Vizuālo piemēru ekrānšāviņi



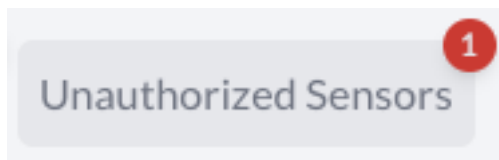
80. attēls. Modālais paziņojuma logs kad sensors mēģina savienoties



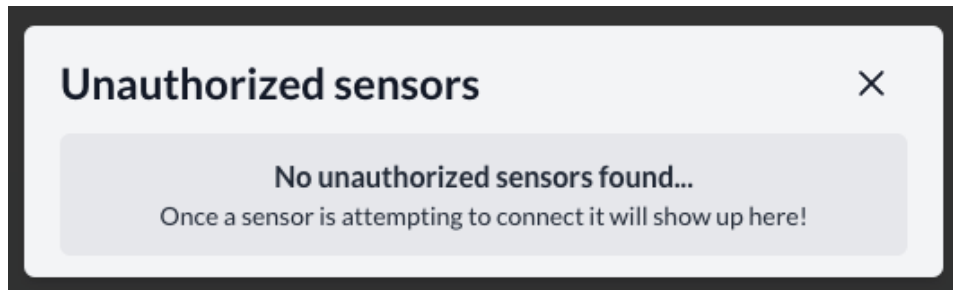
81. attēls. Ar pelēkiem burtiem rakstīts kad notika pēdējais autorizācijas pieprasījums



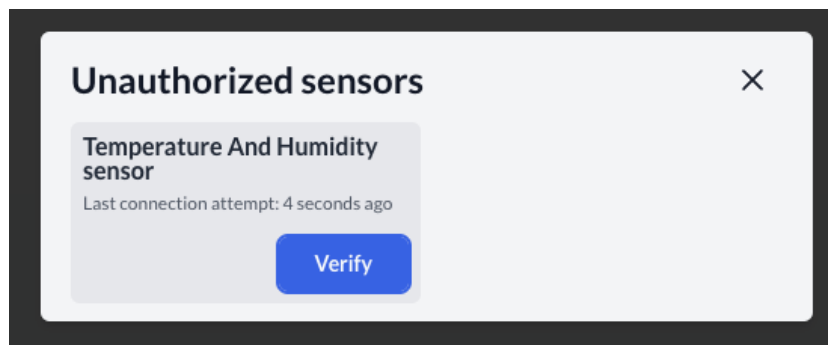
82. attēls. Navigācijas joslā aktīvs iestatījumu skats



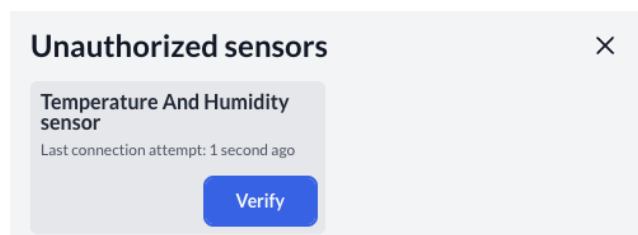
83. attēls. Neautorizēto sensoru poga ar vienu neautorizētu sensoru



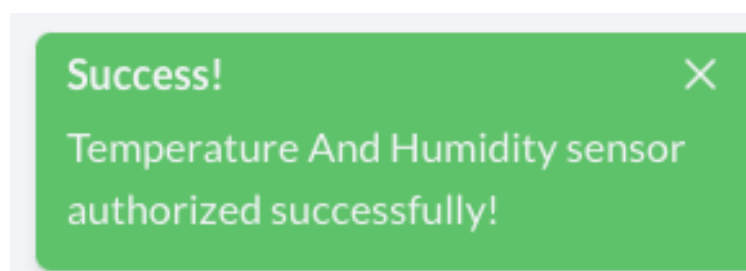
84. attēls. Neautorizēto sensoru saraksta modālais logs, ja nav atrasts neviens sensors



85. attēls. Autorizēto sensoru saraksta modālais logs ar vienu sensoru



86. attēls. Neautorizēto sensoru modālais logs ar vienu sensoru un autorizēšanas pogu



87. attēls. Modālais paziņojuma logs kad sensors tiek veiksmīgi autorizēts

7.17. AUTORIZĒTU SENSORU APSKATE UN DZĒŠANA

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi varētu apskatīt autorizētos sensorus un arī tos dzēst. Tiks paskaidroti sensoru statusi sarakstā (skatīt 4. tabulu). Izpildes soļos tiks arī pievienoti vizuāli piemēri.

4. tabula

Sensoru statusu paskaidrojumi

Statuss	Paskaidrojums
<i>Online</i> (latviešu v. <i>tiešsaistē</i>)	Sensoriem tiek piešķirts statuss <i>Online</i> , ja tas ir veicis kādu datu sūtījumu serverim pēdējo 35 sekunžu laikā. Šis statuss var būt kopā ar <i>Assigned</i> vai <i>Unassigned</i> statusu.
<i>Offline</i> (latviešu v. <i>bezsaitē</i>)	Sensoriem tiek piešķirts statuss <i>Offline</i> , ja tas nav veicis kādu datu sūtījumu 35 sekunžu laikā. Šis statuss var būt kopā ar <i>Assigned</i> vai <i>Unassigned</i> statusu.
<i>Assigned</i> (latviešu v. <i>piesaistīts</i>)	Sensoriem tiek piešķirts statuss <i>Assigned</i> , ja tas ir piesaistīts kādai apakšsekcijai (istabai). Šis statuss var būt kopā ar <i>Online</i> vai <i>Offline</i> statusu.
<i>Unassigned</i> (latviešu v. <i>nepiesaistīts</i>)	Sensoriem tiek piešķirts statuss <i>Unassigned</i> , ja tas nav piesaistīts kādai apakšsekcijai (istabai). Šis statuss var būt kopā ar <i>Online</i> vai <i>Offline</i> statusu.

Izpildes soļi sensoru apskatei

Lai veiksmīgi apskatītu autorizēto sensoru sarakstu, lietotājam nepieciešams:

- atvērt ar navigācijas joslas palīdzību *Settings* (latviešu v. *iestatījumi*) skatu (skatīt 82. attēlā),
- noklikšķināt pogu *Authorized Sensors* (latviešu v. *autorizēti sensori*) (skatīt 88. attēlā).

Rezultāts

Kad tiek nospiesta poga *Authorized Sensors* (latviešu v. *autorizēti sensori*) atvērsies modālais logs ar autorizēto sensoru sarakstu. Ja nav autorizētu sensoru, tad parādīsies ziņojums modālajā logā (skatīt 89. attēlā), savukārt, ja ir autorizēti sensori, tos attēlos sarakstā (skatīt 90. attēlā).

Izpildes soļi sensoru dzēšanai

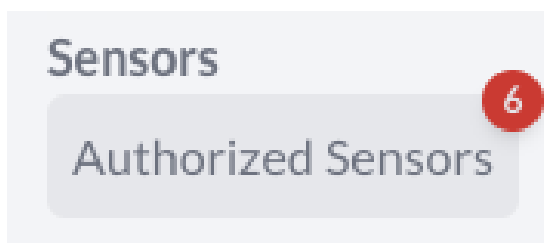
Lai veiksmīgi izdzēstu kādu autorizēto sensoru no saraksta, lietotājam nepieciešams:

- atvērt ar navigācijas joslas palīdzību *Settings* (latviešu v. *iestatījumi*) skatu (skatīt 82. attēlā),
- noklikšķināt pogu *Authorized Sensors* (latviešu v. *autorizēti sensori*) (skatīt 88. attēlā).
- izvēlēties sensoru kuru dzēst, tad nospieš tā sensora taisnstūra labajā augšējā malā trīspunktu ikonu (skatīt 91. attēlā)
- izvēles modālajā logā nospieš pogu *Delete* (latviešu v. *dzēst*) (skatīt 92. attēlā),
- apstiprināšanas modālajā logā nospieš pogu *Delete* (latviešu v. *dzēst*) (skatīt 93. attēlā).

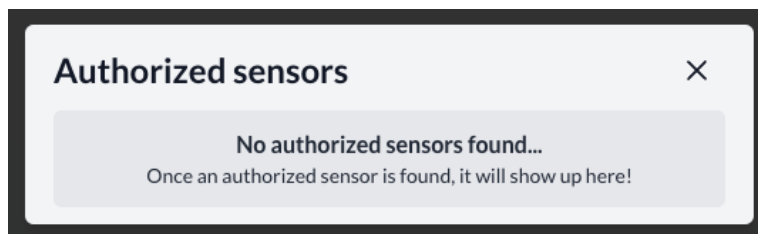
Rezultāts

Kad tiek nospiesta poga *Delete* (latviešu v. *dzēst*), modālais apstiprināšanas logs automātiski aizvērsies, autorizēto sensoru saraksta modālajā logā bez lapas pārlādes pazudīs dzēstais sensors un tīmekļa lietotnes labajā malā parādīsies zaļš modālais logs ar paziņojumu, ka sensors veiksmīgi dzēsts (skatīt 94. attēlā).

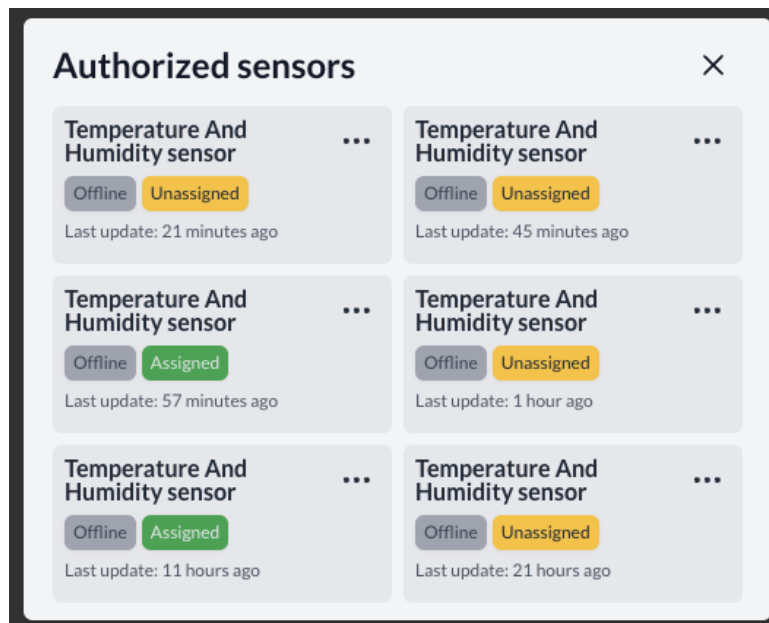
Vizuālo piemēru ekrānšāviņi



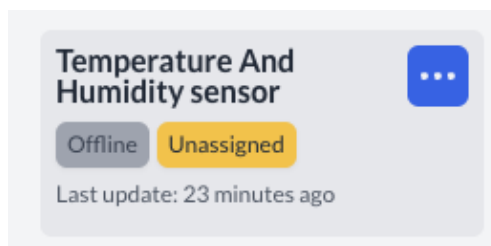
88. attēls. Autorizēto sensoru poga



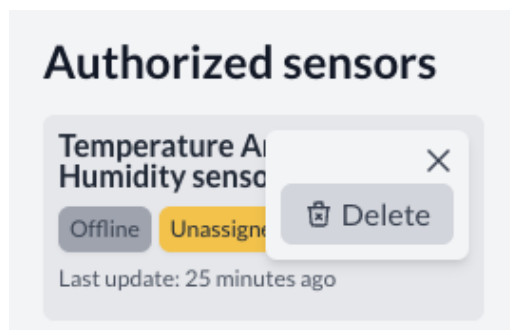
89. attēls. Modālais logs autorizēto sensoru sarakstam, ja nav atrasti sensori



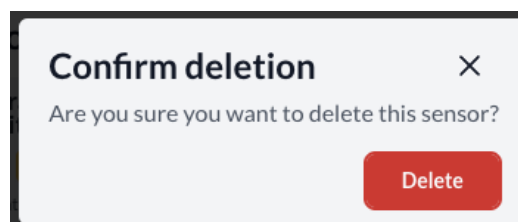
90. attēls. Modālais logs ar autorizēto sensoru sarakstu



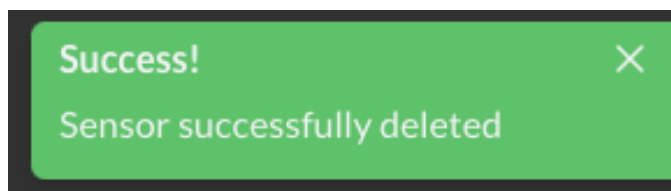
91. attēls. Autorizēts sensors sarakstā ar iezīmētu trīspunktu ikonu



92. attēls. Autorizēta sensora iespēju modālais logs ar iezīmētu dzēšanas pogu



93. attēls. Autorizēta sensora dzēšanas apstiprināšanas modālais logs



94. attēls. Autorizēta sensora veiksmīgas izdzēšanas modālais paziņojums

7.18. LIETOTĀJA IZRAKSTĪŠANĀS NO TĪMEKĻA LIETOTNES

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi spētu izrakstīties no tīmekļa lietotnes. Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

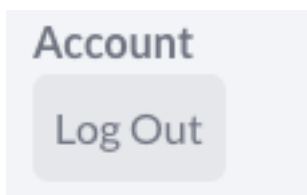
Lai varētu veiksmīgi izrakstīties no tīmekļa lietotnes, lietotājam nepieciešams:

- atvērt ar navigācijas joslas palīdzību *Settings* (latviešu v. *iestatījumi*) skatu (skatīt 82. attēlā),
- noklikšķināt pogu *Log Out* (latviešu v. *izrakstīties*) (skatīt 95. attēlā).
- apstiprināšanas modālajā logā nospiegt pogu *Log Out* (latviešu v. *izrakstīties*) (skatīt 96. attēlā).

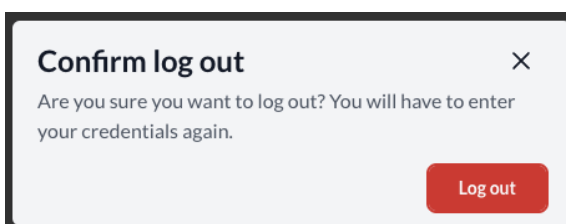
Rezultāti

Kad apstiprināšanas modālajā logā tiek nospiesta poga *Log Out* (latviešu v. *izrakstīties*) lietotājs tiek automātiski pāradresēts uz pieslēgšanās skatu (skatīt 97. attēlā).

Vizuālo piemēru ekrānšāviņi



95. attēls. Izrakstīšanās poga



96. attēls. Izrakstīšanās apstiprināšanas modālais logs

Login

Enter the credentials to Your Smart Home.

Username

Password

Authorize

97. attēls. Pieslēgšanās skats

7.19. TĪMEKĻA LIETOTNES DARBĪBU VĒSTURES ŽURNĀLA APSKATE

Šajā sadaļā tiks aprakstīti detalizēti un secīgi soļi, lai lietotājs veiksmīgi spētu apskatīt lietotnes darbību žurnāla vēsturi (angļu v. *logs*). Izpildes soļos tiks arī pievienoti vizuāli piemēri.

Izpildes soļi

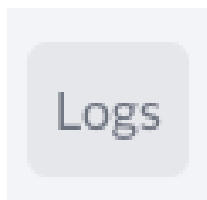
Lai varētu veiksmīgi apskatīt tīmekļa lietotnes darbību vēsturi, lietotājam nepieciešams:

- atvērt ar navigācijas joslas palīdzību *Settings* (latviešu v. *iestatījumi*) skatu (skatīt 82. attēlā),
- noklikšķināt pogu *Logs* (latviešu v. *žurnāls*) (skatīt 98. attēlā).

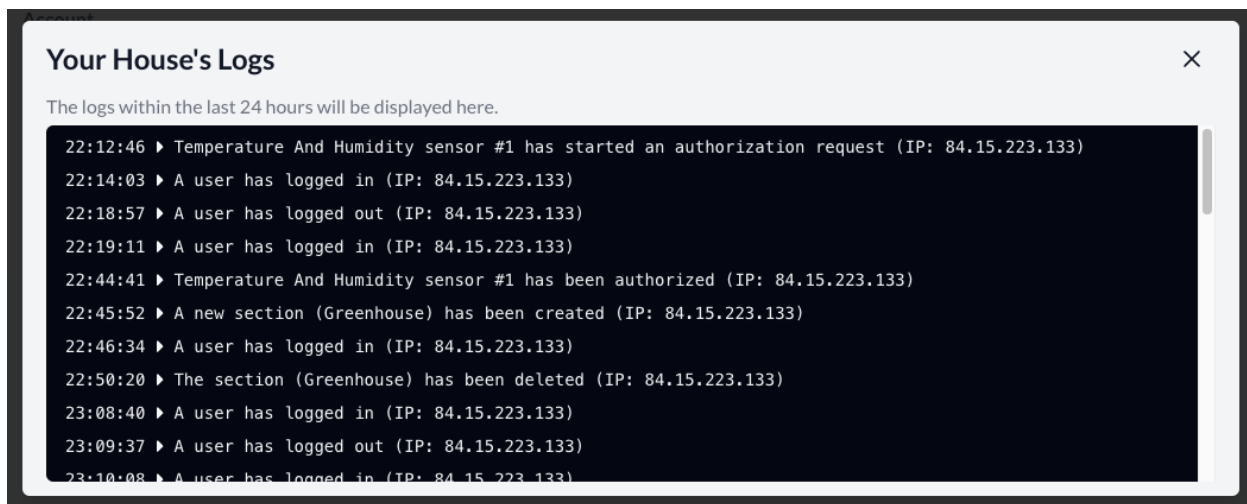
Rezultāti

Kad tiek nospiesta poga noklikšķinātā poga *Logs* (latviešu v. *žurnāls*) lietotāja ekrānā automātiski parādās modālais logs, kur lietotājs var redzēt veiktās darbības pēdējo 24 stundu laikā (skatīt 99. attēlā). Darbību piemēri: pieslēgšanās, izrakstīšanās, datu rediģēšana, datu dzēšana, sensoru autorizācijas pieprasījumi u.c. Pie katras darbības ir ierīces IP adrese no kuras darbība tika veikta.

Vizuālo piemēru ekrānšāviņi



98. attēls. Darbību žurnāla poga



99. attēls. Darbību žurnāla modālais logs ar darbībām

SECINĀJUMI UN PRIEKŠLIKUMI

Kvalifikācijas darba izstrādes rezultātā tika sasniegts mērķis izveidot intuitīvu, modernu un datiem drošu viedmājokļa risinājumu bāzi individuāliem klientiem. Katrs no izvirzītajiem uzdevumiem tika veiksmīgi realizēts, nodrošinot visaptverošu un efektīvu gudrās mājas sistēmu, kas palīdz noteikt energoneefektīvas mājokļa telpas, kā arī nodrošina iespēju apskatīt vēsturiskos telpu mitruma un temperatūras datus.

- **Priekšgalsistēmas izstrāde:** Veiksmīgi izveidota lietotājam draudzīga saskarne, kurā iespējams pārvaldīt sensorus un to datus, izmantojot dažādus grafikus. Saskarne ļauj lietotājiem ērti piekļūt informācijai par mājas vidi un veikt nepieciešamās izmaiņas.
- **Datubāzes arhitektūras izveide:** Izstrādāta un implementēta datubāzes arhitektūra, pamatojoties uz ER diagrammu. Datubāze veiksmīgi uzglabā visu nepieciešamo informāciju par sensoriem, lietotājiem un to datiem, nodrošinot stabilu un drošu datu glabāšanu.
- **Sensoru paraugu izstrāde un kodēšana:** Izveidoti mitruma un temperatūras sensoru paraugi, kā arī izstrādāts to darbības kods. Sensori darbojas precīzi un uzticami, nodrošinot kvalitatīvu datu ievākšanu un nodošanu sistēmai.
- **Aizmugursistēmas izveide:** Izstrādāta aizmugursistēma, kas efektīvi pārvalda datubāzi, uzglabājot sensoru datus, lietotāju pieslēgšanās datus un citus parametrus. Sistēma nodrošina ātru un drošu datu apstrādi un uzglabāšanu.
- **Automatizēta un manuāla testēšana:** Veiksmīgi izstrādāti un veikti automatizēti testi, izmantojot *Laravel* ietvara iebūvēto testēšanas funkcionalitāti. Tika veikta arī manuāla tīmekļa lietotnes testēšana, kas palīdzēja identificēt un novērst radušās problēmas, nodrošinot sistēmas stabilitāti un lietojamību.
- **Problēmu novēršana un labojumu veikšana:** Identificētās problēmas tika veiksmīgi novērstas, veicot nepieciešamos labojumus sistēmā. Tas nodrošināja, ka gala produkts ir funkcionāls, lietotājam draudzīgs un uzticams.

Lai nodrošinātu turpmāku sistēmas attīstību un uzlabošanu, tiek ieteikti šādi priekšlikumi:

- **Integrēt papildus sensoru veidus:** Lai paplašinātu sistēmas funkcionalitāti, būtu lietderīgi integrēt papildus sensoru veidus, piemēram, gaisa kvalitātes, apgaismojuma un kustības sensorus. Tas ļautu vēl precīzāk un detalizētāk uzraudzīt mājas vidi un veikt nepieciešamos pielāgojumus.
- **Izveidot mobilās lietotnes versiju:** Lai nodrošinātu vēl lielāku lietotāju ērtību un pieejamību, būtu ieteicams izstrādāt mobilās lietotnes versiju, kas ļautu lietotājiem pārvaldīt savu gudro māju tieši no viedtālruņa vai planšetdatora.

IZMANTOTĀ LITERATŪRA

1. **Statista.** *Number of users of smarthomes worldwide from 2019 to 2028* [tiešsaistes]. Statista Research department, 2023 [skatīts 08.01.2024]. Pieejams: <https://www.statista.com/forecasts/887613/number-of-smart-homes-in-the-smart-home-market-in-the-world>
2. **BotPenguin.** *The Pros and Cons of Google Home: A Comprehensive Guide* [tiešsaistes]. BotPenguin, 2023 [skatīts 10.01.2024]. Pieejams: <https://botpenguin.com/blogs/the-pros-and-cons-of-google-home-a-comprehensive-guide>
3. **Layne, Jordon.** *20 Pros and Cons of Alexa* [tiešsaistes]. Luxwisp, 2024 [skatīts 19.01.2024]. Pieejams: <https://www.luxwisp.com/pros-and-cons-of-alexa/>
4. **dev.to.** *5 Top Reasons to Use GitHub Actions for Your Next Project* [tiešsaistes]. Davide Benvegnù, 2023 [skatīts 30.05.2024]. Pieejams: <https://dev.to/n3wt0n/5-top-reasons-to-use-github-actions-for-your-next-project-cga>
5. **Shreya Bose.** *Cross Browser Compatibility Issues to Avoid* [tiešsaistes]. BrowserStack, 2023 [skatīts 30.06.2024]. Pieejams: <https://www.browserstack.com/guide/common-cross-browser-compatibility-issues>
6. **Fastfwd.** *Why Choose Laravel for Your Next Web Project* [tiešsaistes]. Fastfwd, 2022 [skatīts 30.05.2024]. Pieejams: <https://www.fastfwd.com/why-choose-laravel-for-your-next-web-project/>
7. **Oleksandr Hutsulyak.** *10 Key Reasons Why You Should Use React for Web Development* [tiešsaistes]. TechMagic, 2023 [skatīts 30.05.2024]. Pieejams: <https://www.techmagic.co/blog/why-we-use-react-js-in-the-development/>
8. **Alex Diaconu.** *Pros and cons of WebSockets* [tiešsaistes]. Ably, 2023 [skatīts 30.05.2024]. Pieejams: <https://ably.com/topic/websockets-pros-cons>
9. **Hao.** *Should you use Docker for local development?* [tiešsaistes]. Hao's learning log, 2021 [skatīts 30.05.2024]. Pieejams: <https://blog.hao.dev/should-you-use-docker-for-local-development>
10. **Richard Makara.** *The Pros and Cons of Using ER Diagrams in Data Modeling* [tiešsaistes]. reconfigured.io, 2022 [skatīts 30.05.2024]. Pieejams: <https://reconfigured.io/blog/pros-and-cons-of-using-er-diagrams-in-data-modeling>
11. **Smartbear.** *The Benefits of Automation Testing* [tiešsaistes]. Smartbear, 2022 [skatīts 30.05.2024]. Pieejams: <https://smartbear.com/learn/automated-testing/>
12. **Visual Paradigm.** *Unveiling the World of Component Diagrams in UML* [tiešsaistes]. Visual Paradigm, 2023 [skatīts 30.05.2024]. Pieejams: <https://guides.visual-paradigm.com/unveiling-the-world-of-component-diagrams-in-uml/>

13. **Peter Koffer.** *Web Development Testing - why is it important?* [tiešsaistes]. Mdevelopers, 2022 [skatīts 30.05.2024]. Pieejams: <https://mdevelopers.com/blog/web-development-testing-why-is-it-important->
14. **Rory Toal.** *A Comprehensive Guide to PHP Programming: What You Need to Know* [tiešsaistes]. codeinstitute, 2022 [skatīts 30.05.2024]. Pieejams: <https://codeinstitute.net/global/blog/what-is-php-programming/>
15. **Glossary.** *What is JavaScript, and why is it important?* [tiešsaistes]. BigCommerce Essentials, 2022 [skatīts 30.05.2024]. Pieejams: <https://www.bigcommerce.com/glossary/javascript/>
16. **Kashyap Vyas.** *8 Major Advantages of Using MySQL* [tiešsaistes]. Datamation, 2023 [skatīts 30.05.2024]. Pieejams: <https://www.datamation.com/storage/8-major-advantages-of-using-mysql/>
17. **Thomas Ryu.** *Should You Use Tailwind CSS?* [tiešsaistes]. Medium, 2023 [skatīts 30.05.2024]. Pieejams: <https://medium.com/@thomas.ryu/should-you-use-tailwind-css-83d519a29448>
18. **Silpa Sasidharan.** *Frontend Architecture | Benefits, Best Practices and The Powerful Ways To Improve Its Design* [tiešsaistes]. ThinkPalm, 2024 [skatīts 30.05.2024]. Pieejams: <https://thinkpalm.com/blogs/frontend-architecture-benefits-and-best-ways-to-improve-its-design/>
19. **Ela Mazurkiewicz.** *Functional vs non-functional requirements* [tiešsaistes]. Inwedo, 2023 [skatīts 30.05.2024]. Pieejams: <https://inwedo.com/blog/functional-and-non-functional-requirements/>
20. **Astari S.** *What Is HTML? Hypertext Markup Language Basics Explained* [tiešsaistes]. Hostinger, 2023 [skatīts 30.05.2024]. Pieejams: <https://www.hostinger.com/tutorials/what-is-html>

PIELIKUMI

1. pielikums. *Arduino* temperatūras un mitruma sensora kods

Lai sensoru varētu pieslēgt pie tīmekļa vietnes un nosūtīt datus, vispirms tiek izveidots savienojums ar *WiFi* tīklu. Kad savienojums ir izveidots, sensors veic autentifikāciju ar serveri, lai pārliecinātos, ka tas ir autorizēts nosūtīt datus. Pēc veiksmīgas autentifikācijas, sensors periodiski nolasīs temperatūras un mitruma datus un nosūtīs tos uz serveri. Ja autentifikācija neizdodas, sensors turpinās mēģināt izveidot savienojumu.

Sensora kods

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

const char* ssid = "TEST"; // WiFi tīkla nosaukums
const char* password = "12345678"; // WiFi tīkla parole

#define DHTPIN D2 // DHT sensors pieslēgts pie D2 pine
#define DHTTYPE DHT11 // DHT sensora tips

DHT_Unified dht(DHTPIN, DHTTYPE); // Iniciē DHT sensoru

// Funkcija, lai ģenerētu unikālu tokenu
String generateToken();

// Mainīgais, kas norāda, vai sensors ir autorizēts
bool authorized = false;

// Unikālais sensora žetons
const String token = generateToken();

// Sensora nosaukums
const String sensor_name = "Temperature And Humidity sensor #1";

// Servera URL autentifikācijai un datu sūtīšanai
const String serverName = "http://api.kantanudarznieciba.lv/api/v1/sensor/handshake";
const String dataServerName = "http://api.kantanudarznieciba.lv/api/v1/sensor/data";

// Laika mainīgie, lai pārvaldītu intervālus
unsigned long previousConnectionMillis = 0;
unsigned long previousDataMillis = 0;
const long connectionInterval = 5000; // 5 sekundes intervāls savienojumam
const long dataInterval = 30000; // 30 sekundežu intervāls datu sūtīšanai

void setup() {
  // Sāk seriālo komunikāciju
  Serial.begin(115200);
  delay(10);

  // Iniciē DHT sensoru
  dht.begin();

  // Pievienojas WiFi tīklam
  Serial.println("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("\nConnected to WiFi");
}
```

```

void loop() {
    unsigned long currentMillis = millis();

    // Mēģina veikt savienojumu, ja nav autorizēts
    if (!authorized && currentMillis - previousConnectionMillis >= connectionInterval) {
        previousConnectionMillis = currentMillis;
        connectionAttempt();
    }
    // Nosūta datus, ja ir autorizēts
    else if (authorized && currentMillis - previousDataMillis >= dataInterval) {
        previousDataMillis = currentMillis;
        sendData();
    }
}

// Funkcija, lai veiktu savienojuma un autorizācijas mēģinājumu ar serveri
void connectionAttempt() {
    if (WiFi.status() == WL_CONNECTED) {
        WiFiClient client;
        HTTPClient http;

        http.setFollowRedirects(HTTPC_FORCE_FOLLOW_REDIRECTS);
        http.begin(client, serverName);

        // Sagatavo datus formai
        String formData = "sensor_name=" + sensor_name + "&sensor_token=" + token;

        // Uztāda saturu tipu galveneī
        http.addHeader("Content-Type", "application/x-www-form-urlencoded");

        // Veic HTTP POST pieprasījumu
        int httpCode = http.POST(formData);

        if (httpCode > 0) {
            Serial.printf("[HTTP] POST request status code: %d\n", httpCode);
            String response = http.getString();
            Serial.println("Response: " + response);

            // Pārbauda, vai sensors ir autorizēts
            if (httpCode == 200) {
                authorized = true;
                Serial.println("Sensor authorized");
            } else {
                Serial.println("Sensor not authorized");
            }
        } else {
            Serial.printf("[HTTP] POST request failed, error: %s\n", http.errorToString(httpCode).c_str());
        }

        http.end();
    } else {
        Serial.println("WiFi not connected");
    }
}

// Funkcija, lai nosūtītu sensora datus uz serveri
void sendData() {
    if (WiFi.status() == WL_CONNECTED) {
        WiFiClient client;
        HTTPClient http;

        http.begin(client, dataServerName);

        sensors_event_t temperatureEvent, humidityEvent;
        dht.temperature().getEvent(&temperatureEvent);
        dht.humidity().getEvent(&humidityEvent);

        float temperature = temperatureEvent.temperature;
        float humidity = humidityEvent.relative_humidity;

        // Pārbauda, vai sensora dati ir derīgi
        if (isnan(temperature)) {
            Serial.println("Failed to read temperature from DHT sensor!");
        } else {

```

```

    Serial.printf("Temperature: %.2f°C\n", temperature);
}

if (isnan(humidity)) {
    Serial.println("Failed to read humidity from DHT sensor!");
} else {
    Serial.printf("Humidity: %.2f%%\n", humidity);
}

if (isnan(temperature) || isnan(humidity)) {
    return;
}

// Sagatavo datus formai
String formData = "sensor_token=" + token + "&temperature=" + String(temperature) + "&humidity=" + String(humidity);

// Uzstāda saturu tipu galveneī
http.addHeader("Content-Type", "application/x-www-form-urlencoded");

// Veic HTTP POST pieprasījumu
int httpCode = http.POST(formData);

if (httpCode > 0) {
    Serial.printf("[HTTP] POST request status code: %d\n", httpCode);
    String response = http.getString();
    Serial.println("Response: " + response);
} else {
    Serial.printf("[HTTP] POST request failed, error: %s\n", http.errorToString(httpCode).c_str());
}

http.end();
} else {
    Serial.println("WiFi not connected");
}
}

// Funkcija, lai ģenerētu unikālu žetonu
String generateToken() {
    const char charset[] = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    const int tokenLength = 30;
    String token = "";

    for (int i = 0; i < tokenLength; i++) {
        int index = random(strlen(charset));
        token += charset[index];
    }

    return token;
}

```

Secīgs koda skaidrojums

1. *WiFi* savienojuma izveide:

- Tiek pievienotas nepieciešamās bibliotēkas un uzstādīti *WiFi* tīkla nosaukumi un parole.
- *setup* funkcijā tiek uzsākta seriālā komunikācija un *DHT* sensora inicializācija.
- Tiek pievienots *WiFi* tīklam un izveidots savienojums.

2. Savienojuma un datu nosūtīšanas pārvaldība:

- *loop* funkcija pārbauda pašreizējo laiku un attiecīgi mēģina veikt savienojuma mēģinājumu, ja sensors nav autorizēts, vai nosūta datus, ja sensors ir autorizēts.

3. Autorizācijas mēģinājums:

- *connectionAttempt* funkcija mēģina savienoties ar serveri, nosūtot *POST* pieprasījumu ar sensora informāciju.
- Ja atbilde no servera ir pozitīva (statusa kods 200), sensors tiek atzīts par autorizētu.

4. Datu nosūtīšana:

- *sendData* funkcija nolasīt temperatūras un mitruma datus no *DHT* sensora.
- Ja sensora dati ir derīgi, tie tiek nosūtīti uz serveri, izmantojot *POST* pieprasījumu.

5. Unikāla žetona ģenerēšana:

- *generateToken* funkcija ģenerē nejaušu žetonu, kas tiek izmantots sensora autentifikācijai.

Šis kods nodrošina, ka sensors var savienoties ar *WiFi* tīklu, veikt autentifikāciju ar serveri un regulāri nosūtīt temperatūras un mitruma datus.