



Diplomski studij

**Informacijska i komunikacijska
tehnologija**

Računarstvo

Telekomunikacije i informatika
Obradba informacija
Računalno inženjerstvo

Ak.g. 2022./2023.

Internet stvari

Kontrola temperature i svjetlosti u prostoru

Dominik Arih

Iwan Ćulumović

Nikola Đokić

Ana Lukenda

Zrinka Pećanić

Petar Stojanović

Sadržaj

1. Uvod	3
2. Uređaji	4
3. IoT platforma	5
4. Korisničke aplikacije	16

1. Uvod

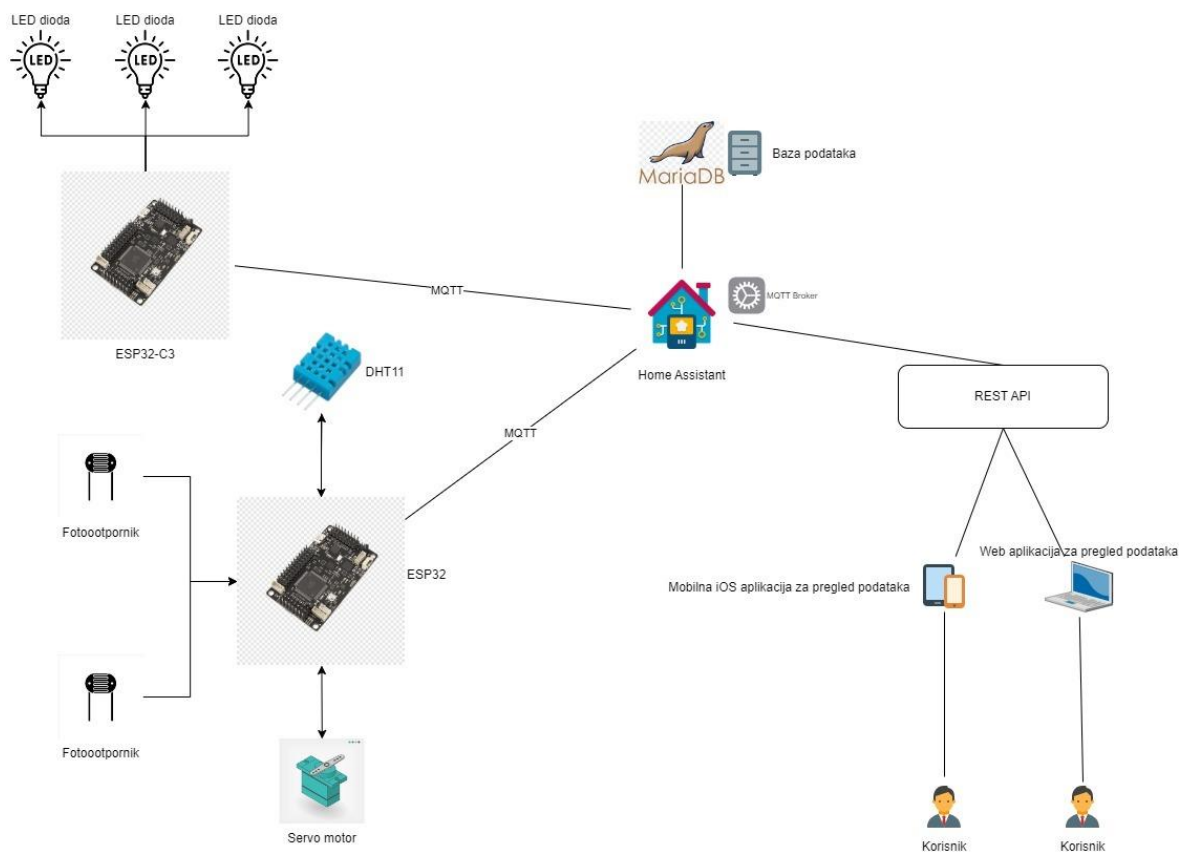
Ideja projekta bila je olakšati kontrolu temperature i svjetlosti u prostoru korištenjem senzorskih i aktuatorskih uređaja preko web i mobilne aplikacije. Uređaji očitavaju vrijednosti temperature, vlage i svjetlosti (DHT11 senzor i svjetlosni otpornici) te prate stanja aktuatora (rolete kontrolirane servomotorom, lampice u prostoru) i to prikazuju u aplikaciji, uz njihovo periodičko ažuriranje svakih 5 sekundi. Korisnik može pregledom vrijednosti donijeti odluku želi li spustiti rolete (kontrola svjetlosti i temperature) i/ili upaliti/ugasiti svjetla (kontrola svjetlosti). Slijede dva primjera uporabe.

Primjer 1: Korisnik otvori aplikaciju na svom mobilnom uređaju i vidi sljedeće vrijednosti: rolete su dignute, temperatura je 29°C, vlaga je 55%, a otpor je malen, dakle svjetlost je jaka, sva su svjetla ugašena. Korisnik zaključuje da je vruće i da je svjetlost u sobi jaka i odabire potpuno spustiti rolete da bi se smanjila temperatura i svjetlost u prostoru. Ta naredba dolazi do uređaja, servomotor se okreće za 180° i rolete su sada spuštene. Korisnik u aplikaciji može vidjeti da su sada rolete spuštene.

Primjer 2: Korisnik otvori web aplikaciju na svom računalu i vidi sljedeće vrijednosti: rolete su dignute, temperatura je 23°C, vlaga je 75%, a otpor je velik, dakle u prostoru je mrak, sva su svjetla upaljena. S obzirom da je noć i korisnik trenutno ne koristi tu sobu, korisnik odabire ugasiti 2 svjetla. U web aplikaciji pri idućem očitavanju korisnik vidi da su sada ta svjetla ugašena.

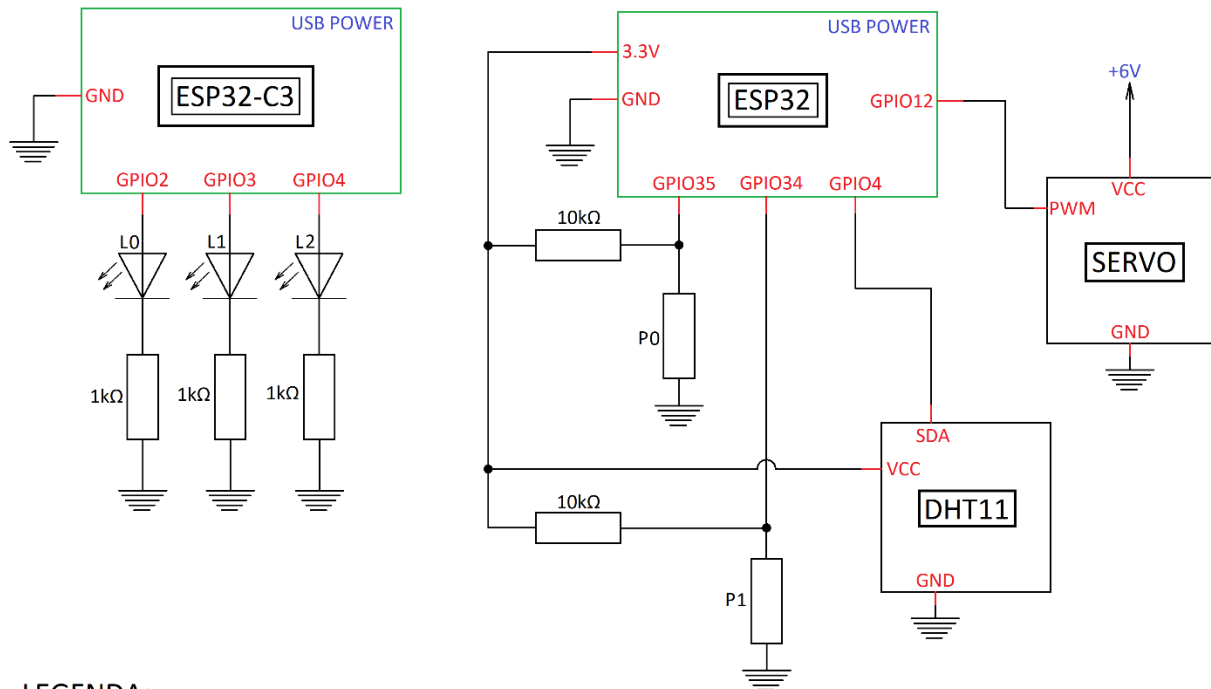
Sličan projekt ovom jest *Smart office* projekt prethodnih generacija na projektu Internet stvari Fakulteta elektrotehnike i računarstva, koji koristi istu platformu (Home Assistant) i senzore (vlaga, temperatura, svjetlost) te slične aktuatore (rolete, pametna utičnica, klima, žarulje).

U nastavku je dana skica našeg rješenja s pripadnim uređajima, platformama i protokolima.



2. Uređaji

Za uređaje su korištene su dvije ESP pločice, tri LED diode, dva svjetlosna otpornika, servomotor i DHT11 senzor za temperaturu i vlagu. Na pločicu ESP32-C3 spojene su tri zelene LED diode (označene s L0, L1, L2) koje služe kao aktuatori za osvjetljenje. Druga pločica jest pločica ESP32 na koju su spojene sljedeće komponente: dva svjetlosna otpornika (označeni s P0, P1) koji služe kao senzori za jačinu svjetlosti u prostoru, DHT11 senzor za očitavanje temperature i vlage i servomotor koji služi kao aktuator za kontroliranje visine roleta. Skica uređaja i njihovog načina povezivanja dana je u nastavku.



LEGENDA:

- > L0, L1, L2 - ZELENE SVJETLEĆE DIODE (LED)
- > P0, P1 - SVJETLOSNI OTPORNIK (10kΩ)

Koristimo Home Assistant i MQTT protokol putem principa objavljivanja i pretplaćivanja za ostvarenje komunikacije između uređaja i platforme.

Definirane su dvije teme: sensors/data i homeassistant/commands.

Home Assistant je pretplaćen na temu sensors/data kako bi očitavao podatke senzora i stanje aktuatora, a objavljuje naredbe za aktuatore na temu homeassistant/commands.

Slijedno tome, obje ESP pločice pretplaćene su na temu homeassistant/commands odakle dobivaju naredbe za aktuatore, a podatke sa senzora i stanje aktuatora periodično objavljuju na temu sensors/data svakih 5 sekundi.

ESP32-C3 šalje poruke o stanju LED dioda u sljedećem obliku na temu sensors/data:

- `{"LED0":"ON"}` ako je LED dioda L0 upaljena ili `{"LED0":"OFF"}` ako je ugašena
- `{"LED1":"ON|OFF"}` za diodu L1
- `{"LED2":"ON|OFF"}` za diodu L2

ESP32 šalje poruke o stanju (kutu) roleta i očitavanja sa svjetlosnih otpornika i DHT11 senzora u sljedećem obliku na temu sensors/data:

- `{"SERVO":{"broj"},""}` (kut servomotora/roleta, raspon 0-180°)
- `{"PHOTORESISTOR0":{"broj"},""}`, `{"PHOTORESISTOR1":{"broj"},""}` (otpor na otpornicima – veći otpor -> mračnije)
- `{"HUMIDITY":{"broj"},""}` (vlaga u %)
- `{"TEMPERATURE":{"broj"},""}` (temperatura u °C)

Home Assistant šalje sljedeće naredbe na temu homeassistant/commands (koje obje ESP pločice primaju i parsiraju te izvršavaju ili ignoriraju - ovisno kojoj je naredba namijenjena):

- `LED0:{ON|OFF}` (paljenje/gašenje svjetla L0)
- `LED1:{ON|OFF}` (paljenje/gašenje svjetla L1)
- `LED2:{ON|OFF}` (paljenje/gašenje svjetla L2)
- `SERVO:{broj}°` (postavljanje servomotora/rolete na određeni kut)

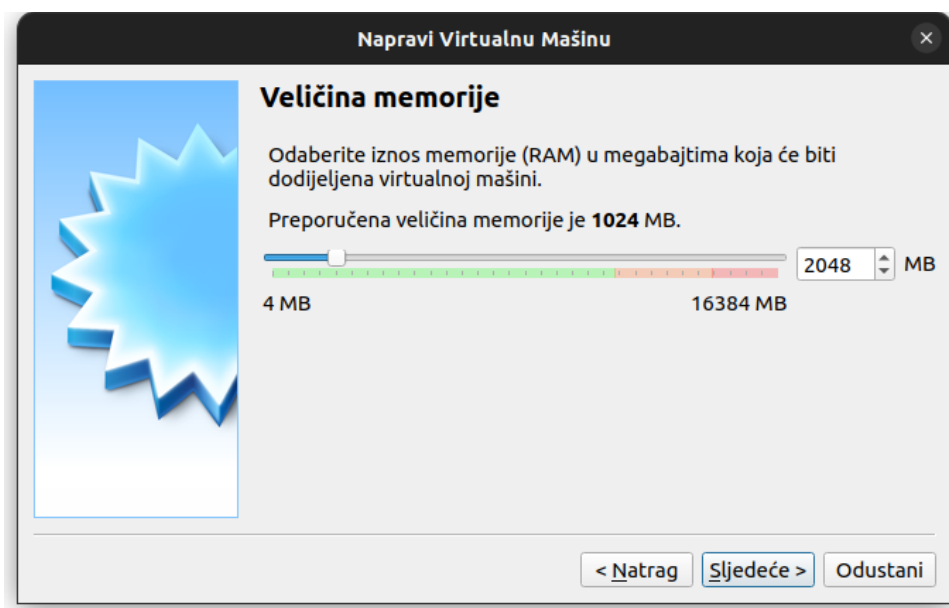
3. IoT platforma

Ovaj projekt je koristio lokalnu besplatnu IoT platformu Home Assistant. Home Assistant je platforma otvorenog koda pisana u Python-u. Podržava različite protokole i uređaje kao primjerice Zigbee, Z-Wave, Bluetooth, MQTT i slično. Pogodan je za instalirati na računala poput Raspberry Pi-a. Bitna sposobnost Home Assistanta je integracija i povezivanje različitih uređaja u mrežu. Home Assistant-om je moguće izraditi razne automatizacije, praćenje vrijednosti senzora te spremanje tih podataka u bazu podataka. Postavke se mogu postavljati grafičkim sučeljem (web browser) ali za bolju kontrolu i većim opsegom postavki je najbolje pisati konfiguraciju u YAML datotekama. Home Assistant također nudi niz proširenja (addon-ova) kojima se može proširiti funkcionalnost cjelokupnog sustava. Dodatno, omogućuje integraciju i s drugim platformama poput Google Assistant-a, IFTTT-om, Amazon Alex-om ... Iz ovog se može zaključiti da je Home Assistant moćan softver za pametne domove koji omogućuje centralizirano upravljanje i automatiziranje doma, integrirajući različite uređaje i servise te pružajući kontrolu i praćenje putem jednostavnog sučelja.

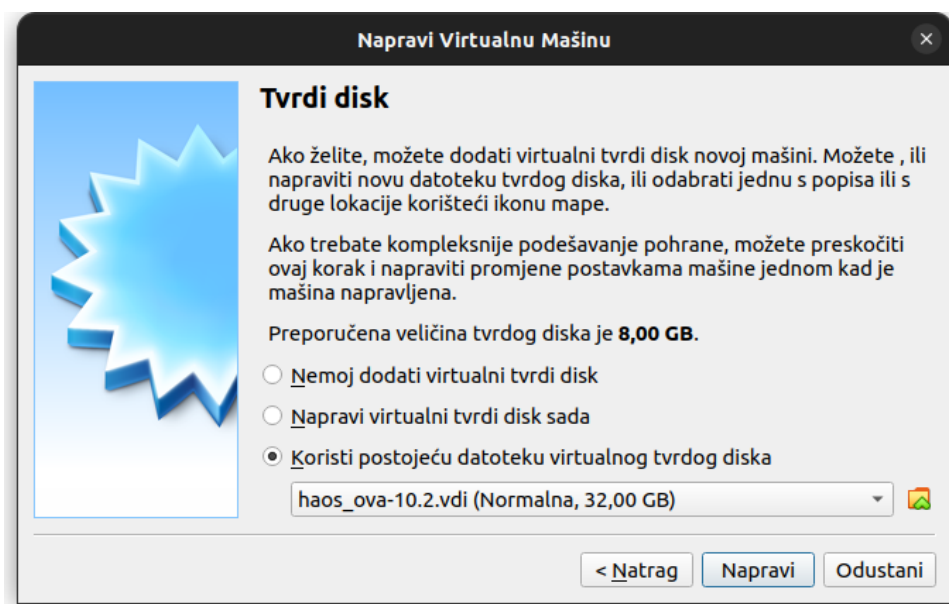
a. Instalacija i prvo pokretanje

U svrhu projekta, HomeAssistant se pokretao na virtualnom stroju koji se pokretao na Ubuntu Desktop-u. Potrebna slika diska (za Virtual Box) na kojem je lagana Linux distribucija HaasOS na kojem se pokreće Home Assistant može se preuzeti sa poveznice: https://github.com/home-assistant/operating-system/releases/download/10.2/haos_ova-10.2.vdi.zip . Arhivu je potrebno raspakirati. Ova instalacija će se koristiti za Virtual Box - alat za virtualizaciju. Otvaranjem alata Virtual Box, potrebno je odabrati opciju "Nova" kojem će se stvoriti Virtualni stroj. Zatim je potrebno napisati ime virtualnog stroja te odabrati vrstu Linux.

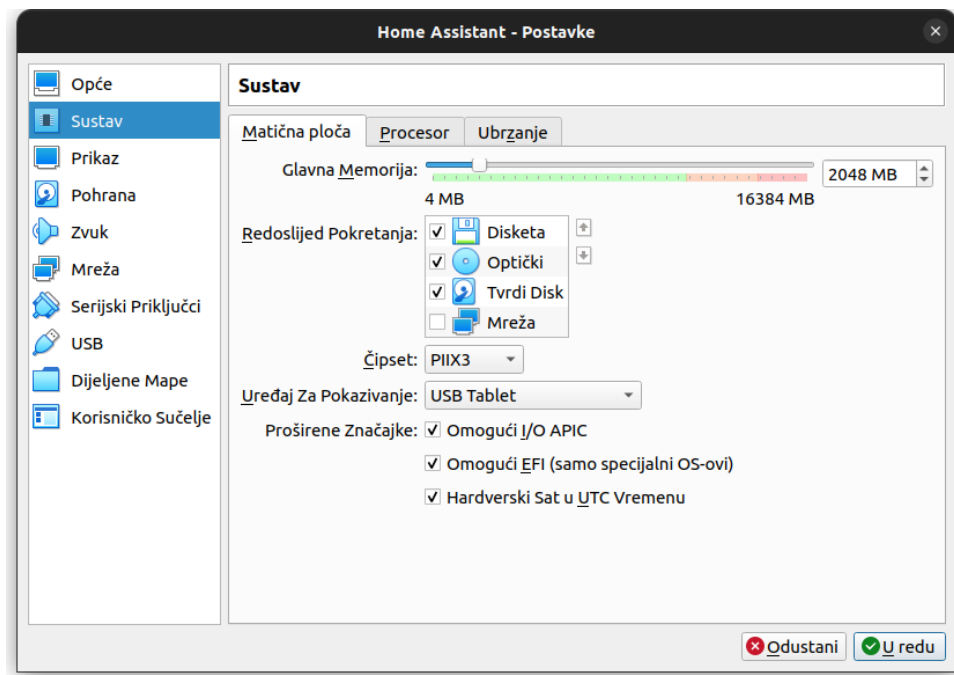
Za normalan rad Home Assistant zahtjeva barem 2 GB radne memorije, stoga je potrebno odabrati 2 GB na sljedećem prozoru:



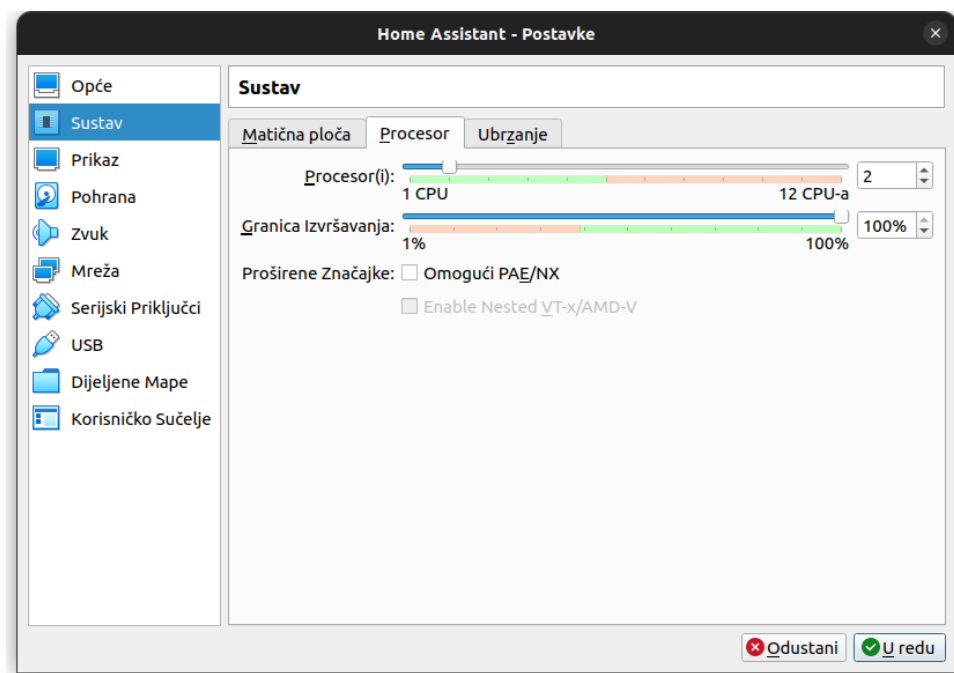
Na sljedećoj stavci potrebno je odabrati sliku diska koja se prethodno otpakirala (nastavak vdi):



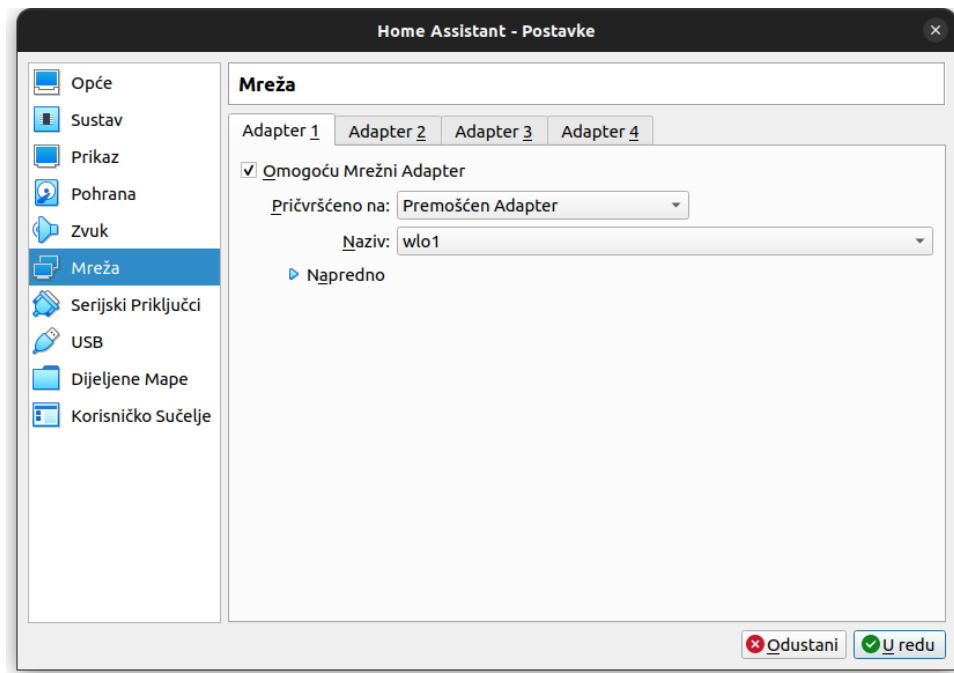
Nakon stvaranja virtualnog stroja, potrebno je postaviti konfiguraciju klikom na novostvoreni virtualni stroj i odabrati opciju "Postavke" te na prvoj kartici opcije "Sustav" omogućiti EFI:



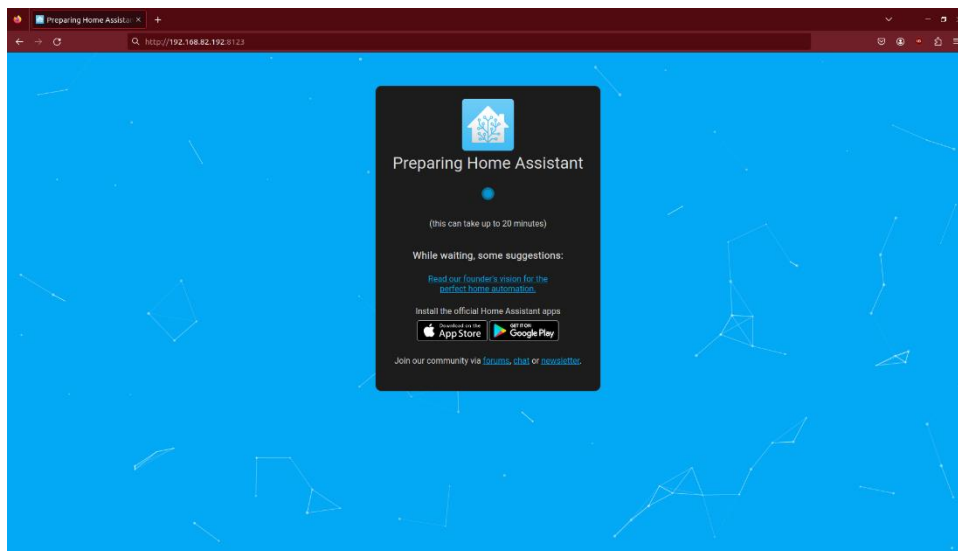
Zatim je potrebno dodijeliti virtualnom stroju dvije jezgre domaćinovog operacijskog sustava:



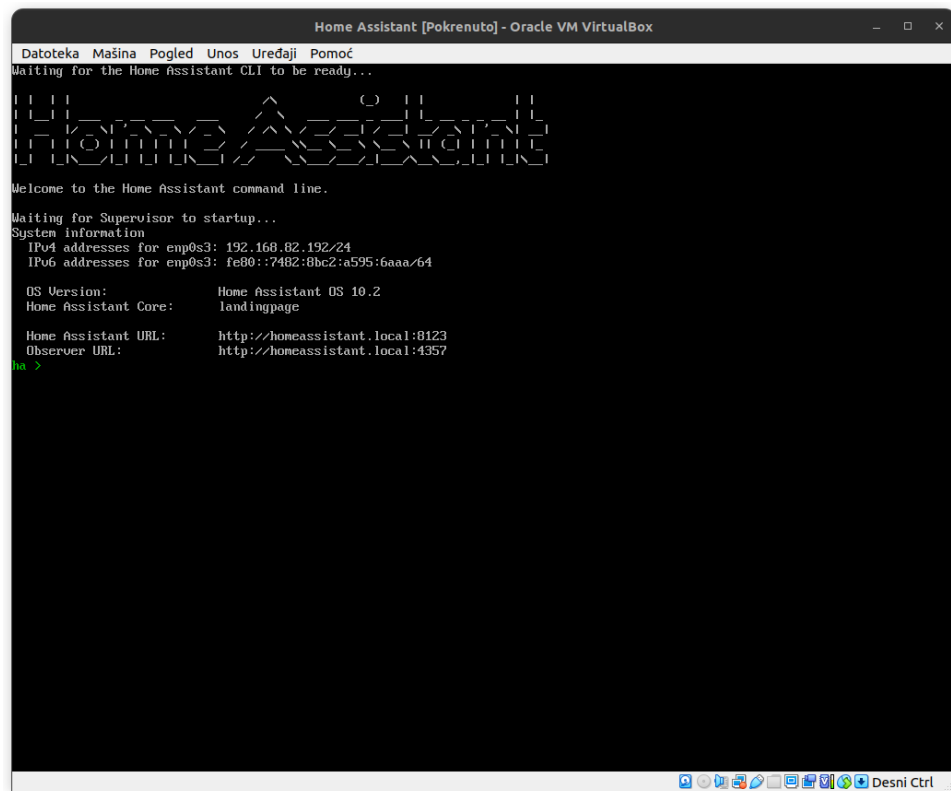
Naposlijetku je potrebno postaviti virtualni mrežni most tako da se virtualni stroj vidi na LAN mreži jer se u ovom slučaju neće koristiti translacija adresa nego će svaki stroj imati vlastitu IP adresu (kao da dva neovisna računala spojena na istu mrežu):



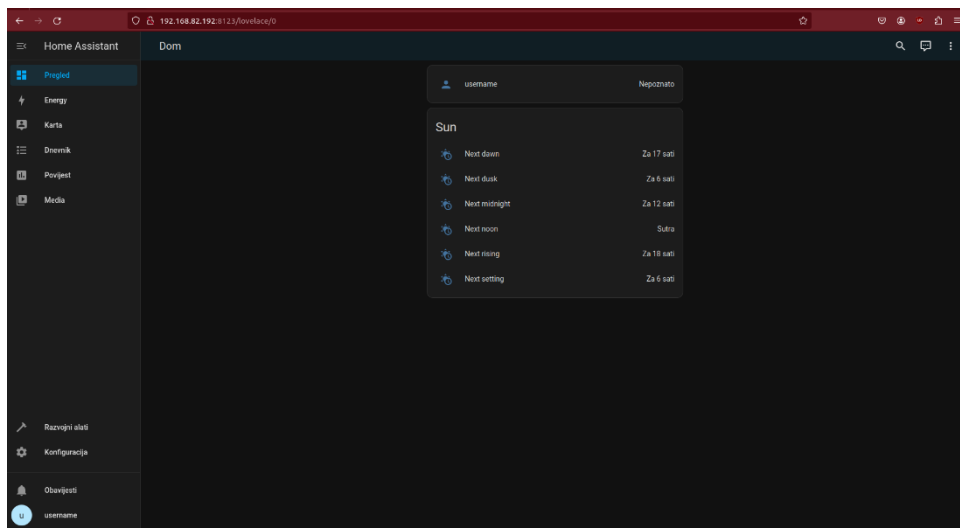
Nakon klika na "U redu" potrebno je pokrenuti virtualni stroj. Nakon pokretanja virtualnog stroja otvorit će se novi prozor koji će sadržavati IP adresu virtualnog stroja. Za pristup Home Assistant-u potrebno je IP adresu upisati u Web preglednik te joj nadodati port 8123 odnosno, URL je oblika: "http://<IP adresa>:8123" odnosno:



IP adresa se saznaje iz prozora koji se otvorio pokretanjem Home Assistant-a:



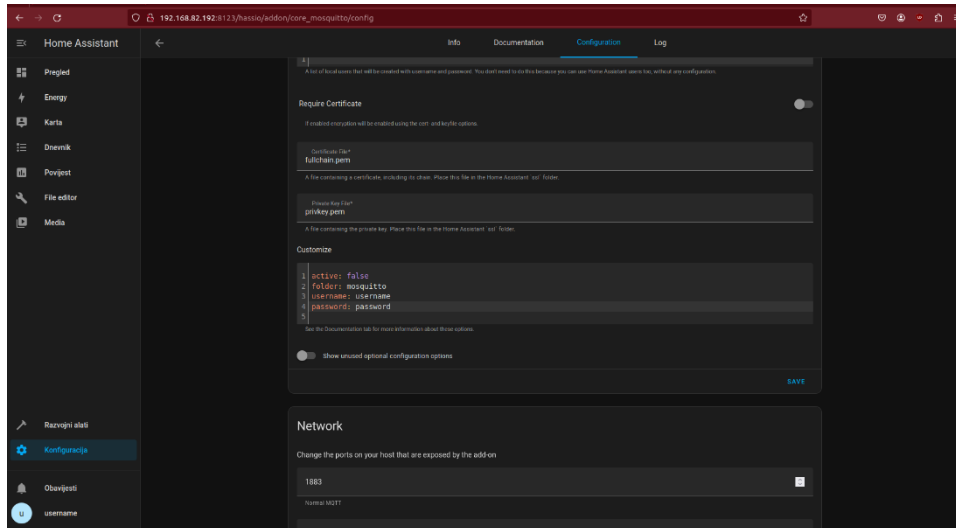
Nakon nekog vremena doći će obrazac za ispunjavanje podataka; korisničkog imena i lozinke te raznih drugih atributa. Obavezne attribute potrebno je ispuniti te se naposljetku dobiva sljedeći prozor:



b. Instalacija proširenja

– MQTT posrednik

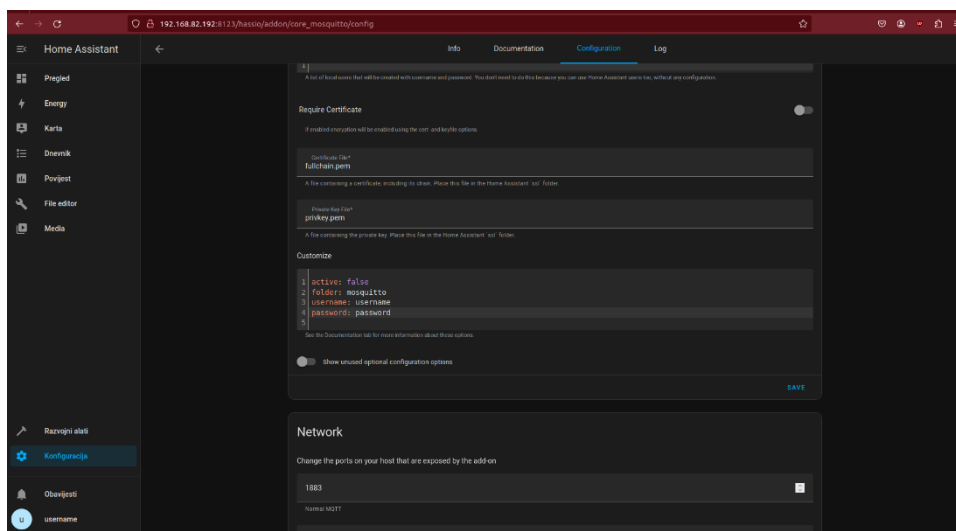
Za instalaciju proširenja MQTT posrednika potrebno je odabrati opciju "Konfiguracija" pa "Add-ons" te "Add-on store". U tražilicu je potrebno upisati "Mosquitto broker" te kliknuti na "Install". Nakon instalacije, potrebno je pod stavkom "Configuration" nadodati "username: username" i "password: password" :



Potrebno je pokrenuti broker odabirom opcije "Start". Zatim je opet potrebno kliknuti na "Konfiguracija" i odabrati "Devices & Services" te pod stavkom "Mosquitto broker" odabrati "KONFIGURIRAJ".

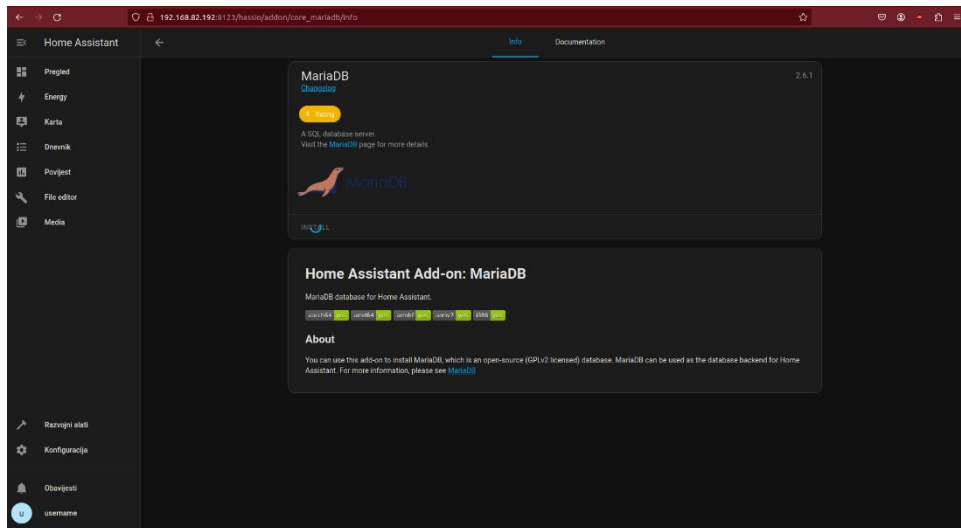
– File editor

Kako bi se uopće mogle uređivati YAML datoteke, potreban je uređivač datoteka File Editor. On je također dostupan kao proširenje. Tražilicom je potrebno pretražiti "Add-on store" te odabrati "File Editor". Potrebno je odabrati "Show in sidebar" kako bi mu se moglo pristupiti te ga je potrebno pokrenuti:



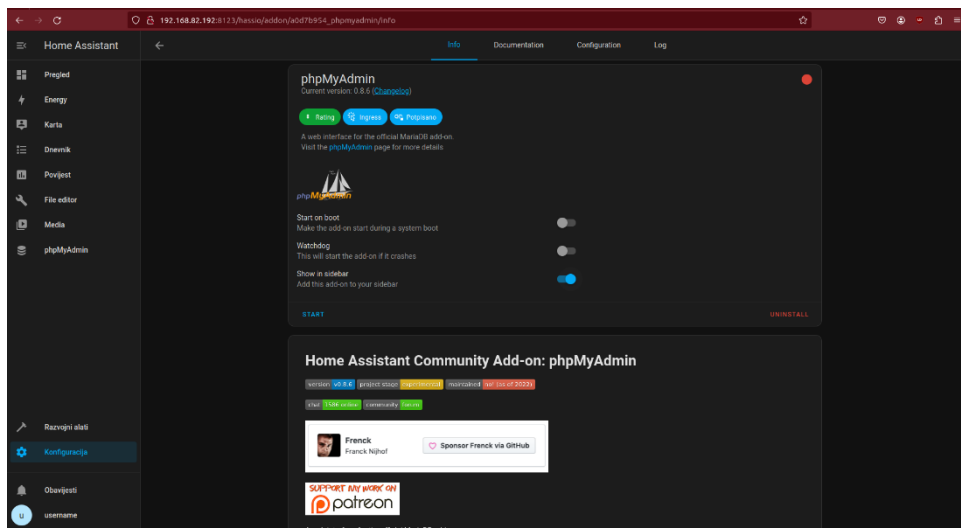
– MariaDB

Za spremanje podataka očitanih sa senzora potrebna je neka baza podataka. Home Assistant nudi već ugrađeni upravitelj bazama podataka SQLite. SQLite je jednostavan i ne nudi puno mogućnosti. Kako bi se lakše upravljalo većom količinom podataka, nudi se upravitelj sustavama baza podataka MariaDB. Instalacija se vrši preko "Add-on store"-a. Nakon instalacije, potrebno je konfigurirati bazu podataka pod karticom "Configuration" te na mjestu password: umjesto "null" postaviti "password".



– phpAdmin

Kako bi se uopće moglo upravljati bazom podataka potreban je neki administracijski alat. On omogućuje stvaranje i brisanje baza podataka, tablica, shema i slično. No u ovom slučaju on će se koristiti, ako bude potrebno, za debugiranje. Instalacija se vrši preko "Add-on store"-a. Nakon instalacije također ga je potrebno postaviti na bočni panel (opcija "Show in sidebar").



Nakon instalacije svih proširenja potrebno je resetirati Home Assistant klikom na "Razvojni alati" te "PONOVO POKRETANJE".

c. Postavke senzora, spremanje i čitanje baze podataka

Instalacijom File editor-a omogućen je pristup bitnim konfiguracijskim datotekama. Prva konfiguracijska datoteka koja je bitna je configuration.yaml. U njoj se postavljaju opće postavke, primjerice konfiguracije same platforme ili konfiguracija entiteta. U prethodnim poglavljima je naglašeno da se komunikacija odvija pomoću protokola MQTT. Oba ESP mikroupravljača koriste MQTT za objavljivanje stanja mikroracunala. U sustavu postoje sveukupno osam očitavanja: tri stanja LED dioda, stanje servo motora, 2 senzora za svjetlinu te temperatura i vlažnost. Kako bi se ti podatci mogli zasebno slagati u bazi podataka, potrebno je stvoriti vlastite entitete odnosno senzore. Također, potrebno je naglasiti da se podatci spremaju u vanjskoj bazi podataka (MariaDB), ne već ugrađenoj koja dolazi sa HomeAssistant-om (SQLite). Datoteka configuration.yaml bi trebala u konačnici izgledati ovako:

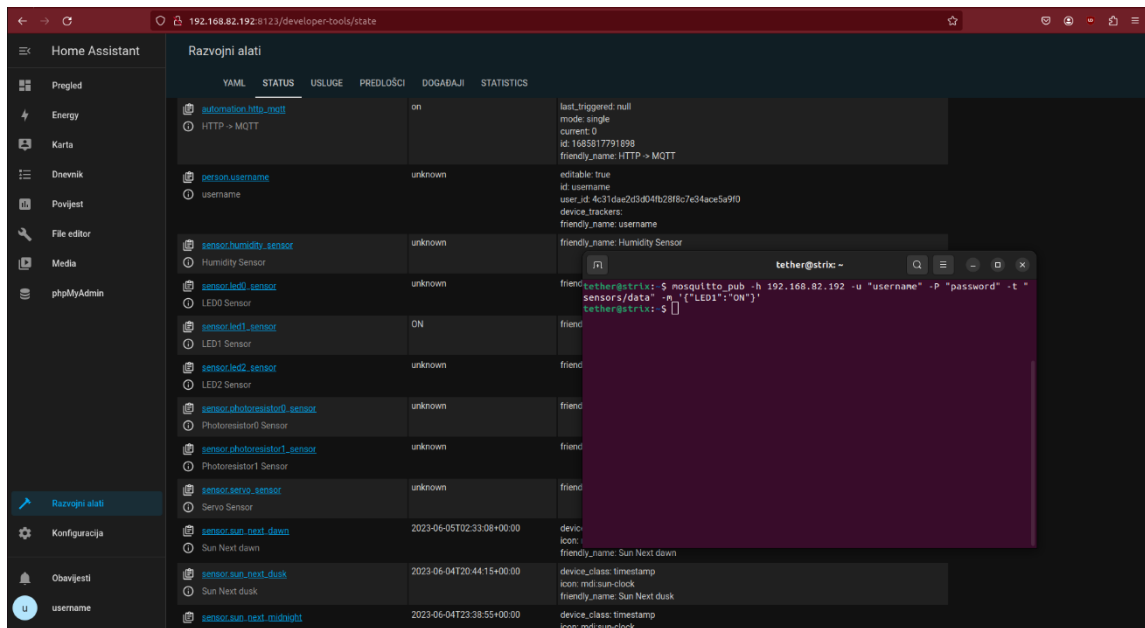
```
/config/configuration.yaml
1 |
2 | # Loads default set of integrations. Do not remove.
3 | default_config:
4 |
5 | # Load frontend themes from the themes folder
6 | frontend:
7 |   themes: !include_dir_merge_named themes
8 |
9 | # Text to speech
10 | tts:
11 |   platform: google_translate
12 |
13 | automation: !include automations.yaml
14 | script: !include scripts.yaml
15 | scene: !include scenes.yaml
16 |
17 | http:
18 |   cors_allowed_origins:
19 |     - '*'
20 |
21 | recorder:
22 |   db_url: mysql://homeassistant:password@core-mariadb/homeassistant?charset=utf8mb4
23 |   include:
24 |     entities:
25 |       - sensor.photoresistor0_sensor
26 |       - sensor.photoresistor1_sensor
27 |       - sensor.temperature_sensor
28 |       - sensor.humidity_sensor
29 |       - sensor.servo_sensor
30 |       - sensor.led0_sensor
31 |       - sensor.led1_sensor
32 |       - sensor.led2_sensor
33 |
34 | mqtt:
35 |   sensor:
36 |     - name: "Photoresistor0 Sensor"
37 |       state_topic: "sensors/data"
38 |       value_template: "{{ value_json.PHOTORESISTOR0 | default(states('sensor.photoresistor0_sensor')) }}"
39 |     - name: "Photoresistor1 Sensor"
40 |       state_topic: "sensors/data"
41 |       value_template: "{{ value_json.PHOTORESISTOR1 | default(states('sensor.photoresistor1_sensor')) }}"
42 |     - name: "Temperature Sensor"
43 |       state_topic: "sensors/data"
44 |       value_template: "{{ value_json.TEMPERATURE | default(states('sensor.temperature_sensor')) }}"
45 |     - name: "Humidity Sensor"
46 |       state_topic: "sensors/data"
47 |       value_template: "{{ value_json.HUMIDITY | default(states('sensor.humidity_sensor')) }}"
48 |     - name: "Servo Sensor"
49 |       state_topic: "sensors/data"
50 |       value_template: "{{ value_json.SERVO | default(states('sensor.servo_sensor')) }}"
51 |     - name: "LED0 Sensor"
52 |       state_topic: "sensors/data"
53 |       value_template: "{{ value_json.LED0 | default(states('sensor.led0_sensor')) }}"
54 |     - name: "LED1 Sensor"
55 |       state_topic: "sensors/data"
56 |       value_template: "{{ value_json.LED1 | default(states('sensor.led1_sensor')) }}"
57 |     - name: "LED2 Sensor"
58 |       state_topic: "sensors/data"
59 |       value_template: "{{ value_json.LED2 | default(states('sensor.led2_sensor')) }}"
```

Pod "mqtt" definira se 8 senzora te svi prisluškuju istu MQTT temu. Ovisno o tome jeli u pridošloj poruci došao JSON sa odgovarajućim ključem, stanje senzora će se ažurirati, inače stanje ostaje isto.

Pod "recorder" se definira koja baza podataka će se koristiti za spremanje povijesti podataka. Uključeno je spremanje podataka za svih osam senzora.

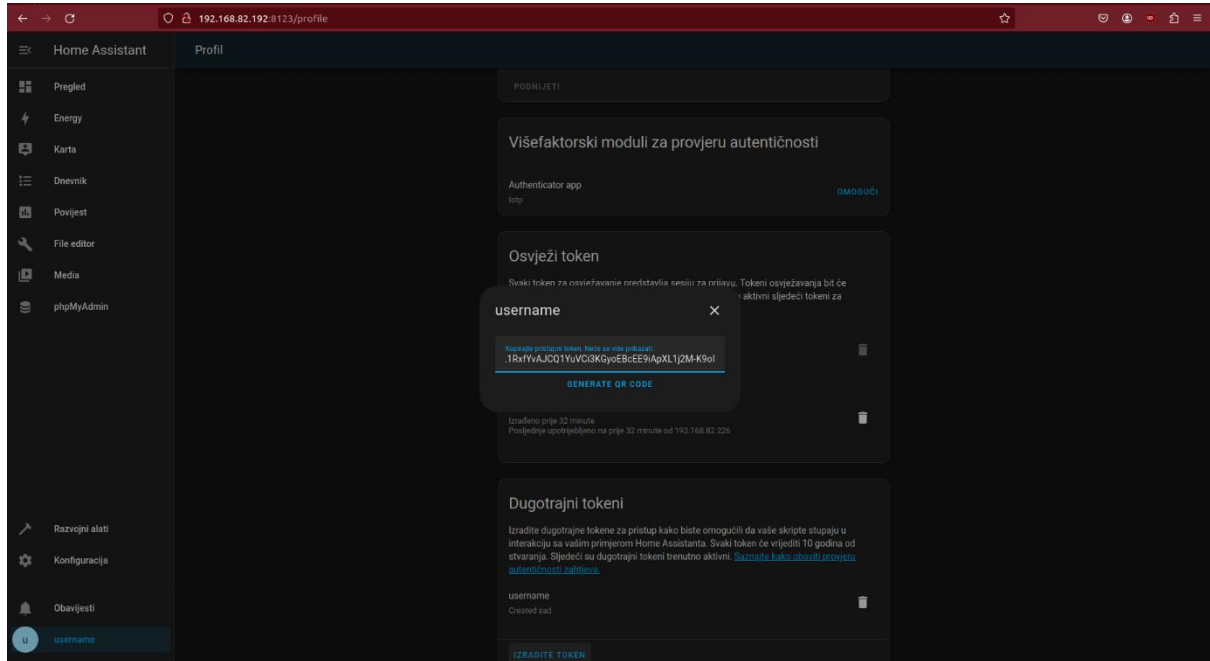
Pod "http" se definira koje IP adrese imaju pristup resursima ovog sustava (resursima Home Assistant-a). U ovom slučaju svi koji su spojeni na lokalnu mrežu smiju dohvaćati resurse odnosno web stranice.

Senzori će naposljetku biti stvoreni te im se promjena stanja može ispitati objavljivanjem poruka na temu sensors/data:



d. Posluživanje vlastitih web stranica

Kako bi se omogućilo posluživanje web stranica, potrebno je u config direktoriju stvoriti direktorij www i u njemu stvoriti datoteke koje sadrže potreban kod (HTML, CSS, JavaScript). Pristup stranicama odvija se pomoću poveznice: `http://<IP adresa>:8123/local/<ime datoteke>`. Međutim, HTTP zahtjev u svojem zaglavlju mora imati dugotrajni token radi autentifikacije. Njega se može generirati klikom na profile (korisničko ime) te "IZRADITE TOKEN":



e. Prosljeđivanje POST zahtjeva na MQTT

Za prosljeđivanje naredbi sa web preglednika na MQTT potreban je okidač i akcija. Okidač bi trebao biti POST zahtjev koji nosi JSON objekt koji sadrži što treba objaviti na temu homeassistant/commands. Za ovakve automatizacije potrebno je urediti datoteku automations.yaml sljedećim sadržajem:

```
/config/automations.yaml
1 - id: '1685817791898'
2   alias: HTTP -> MQTT
3   description: HTTP_POST -> MQTT_PUBLISH
4   trigger:
5     - platform: webhook
6     allowed_methods:
7       - POST
8       - PUT
9     local_only: true
10    webhook_id: command
11    condition: []
12    action:
13      - service: mqtt.publish
14        data:
15          qos: 0
16          retain: false
17          topic: homeassistant/commands
18          payload: '{{ trigger.json.message }}'
19    mode: single
20
```

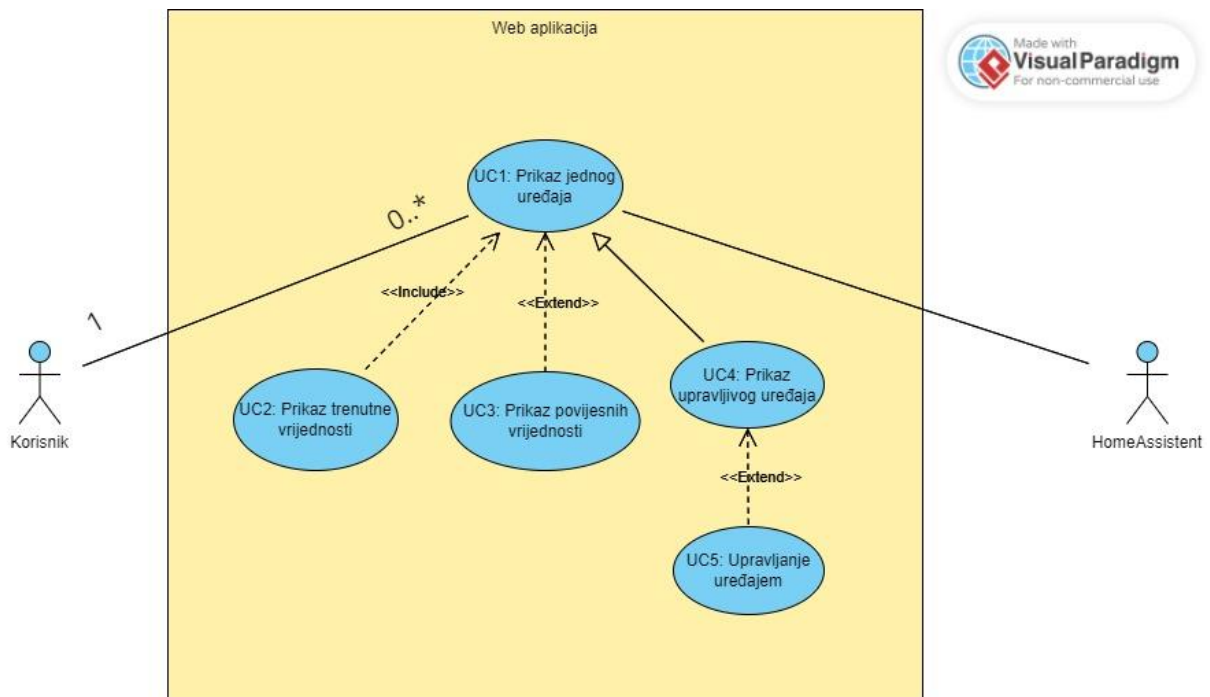
Za okidač se koristi WebHook. WebHook je usluga koja omogućuje "notifikacije" od jednog sustava na drugi. Kad se napravi POST zahtjev na URL: `http://<IP adresa>:8123/api/webhook/command`, okidač će pokrenuti uslugu `mqtt.publish` koja će naposljetku objaviti vrijednost ključa `message` u poslanoj JSON poruci na temu `homeassistant/commands`.

4. Korisničke aplikacije

Razvijene su web i iOS mobilna aplikacija za upravljanje sustavom.

Web aplikacija

U nastavku je dan use-case dijagram za web aplikaciju:



Slijedi izgled početne stranice:



U nastavku je dan prikaz stranice s povijesnim mjerenjima:

History of photoresistor0	
Friendly Name: Photoresistor0 Sensor State: 714 Last Changed: 2023-06-04T15:15:34.647339+00:00	
Friendly Name: Photoresistor0 Sensor State: 713 Last Changed: 2023-06-04T15:15:29.628828+00:00	
Friendly Name: Photoresistor0 Sensor State: 727 Last Changed: 2023-06-04T15:15:24.559964+00:00	
Friendly Name: Photoresistor0 Sensor State: 700 Last Changed: 2023-06-04T15:15:19.531680+00:00	
Friendly Name: Photoresistor0 Sensor State: 718 Last Changed: 2023-06-04T15:15:14.499026+00:00	
Friendly Name: Photoresistor0 Sensor State: 716 Last Changed: 2023-06-04T15:15:09.484040+00:00	
Friendly Name: Photoresistor0 Sensor State: 714 Last Changed: 2023-06-04T15:15:04.549065+00:00	

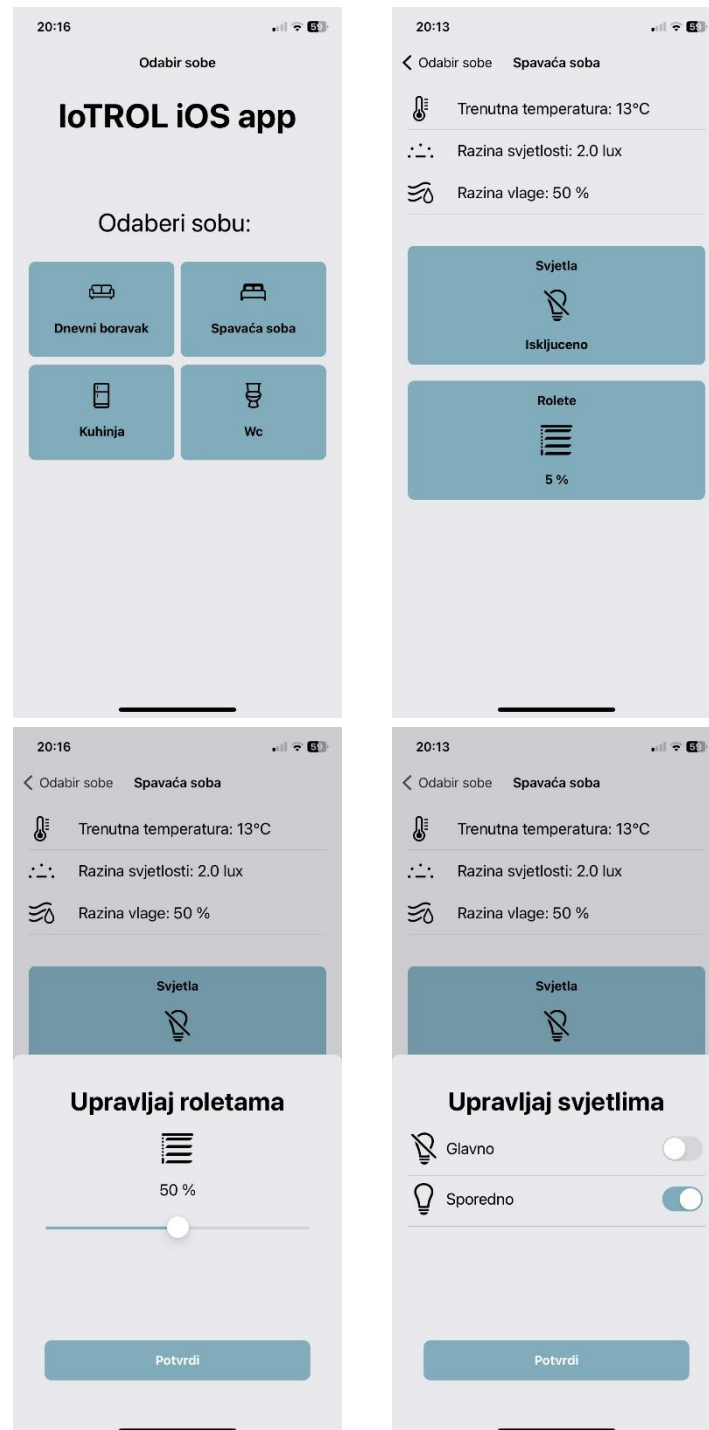
Napravljena je jednostavna aplikacija s razvijenim frontendom, backend nije razvijen. To je nedostatak iz razloga što se broj i vrsta uređaja ne može dinamički učitavati i prikazivati već je statički unesen i svaki novi uređaj se mora ručno unositi. Funkcionalnosti su ostvarene pomoću JavaScripta bez korištenja dodatnih tehnologija. Svi potrebni resursi pohranjeni su u Home Assistant konfiguracijskom direktoriju te se stranici može pristupiti na adresi `http://<IP adresa>:<PORT>/local/homepage.html`, pri čemu se upisuje IP adresa računala na kojem se nalazi Home Assistant, a PORT 8123. Za komunikaciju s Home Assistantom korišten je REST API.

Na početnoj stranici prikazuju se svi dostupni uređaji. Za svaki uređaj prikazuje se trenutno stanje, za senzore je to očitana vrijednost, a za ostale uređaje stanje aktivnosti. Pri prvom učitavanju dohvaća se trenutna vrijednost te se postavlja da se nova vrijednost učitava svake dvije sekunde. Za uređaje kojima je moguće upravljati nudi se mogućnost upravljanja, a za sve se uređaje nudi i prikaz prethodnih mjerenja. Roletama se upravlja drukčije nego ostalim uređajima, postavljen je klizni izbornik s mogućnošću spuštanja roleta od 0% do 100 % te je odabranu vrijednost potrebno predati pritiskom na Submit gumb kako bi se akcija primijenila. Pritiskom na gumba za prikaz povijesnih mjerenja nekog uređaja, odlazi se na novu stranicu na kojoj se prikazuje zadnjih 10 mjerenja iz baze.

Komunikacija s Home Assistantom radi dohvaćanja podataka i upravljanja uređajima ostvarena je pomoću REST API-ja, podaci se dohvaćaju slanjem GET zahtjeva na adresu `http://<IP adresa>:<PORT>/api/states/<entity_id>`, pri čemu je IP adresa adresa Home Assistant, PORT 8123 (defaultni port za slanje REST zahtjeva), a `entity_id` oznaka za traženi uređaj, definirana u `configuration.yaml` datoteci Home Assistant. Za uspješan pristup podacima potrebna je i autentifikacija te se stoga u zaglavlju zahtjeva prosljeđuje pristupni token generiran u Home Assistantu. Za upravljanje uređajima šalje se POST poruka na adresu `http://<IP adresa>:<PORT>/api/webhook/command`, a u poruci se prosljeđuje oznaka uređaja i željena akcija. Razmjenjivane poruke u JSON su formatu. Pri dohvaćanju povijesnih podataka šalje se GET zahtjev na `http://<IP adresa>:<PORT>/api/history`. Za slanje REST zahtjeva JavaScriptom korišten je FETCH API. Stilovi i funkcije potrebne za ispravno učitavanje `homepage.html` datoteke definirani su u `style.css` i `script.js` datotekama, a za učitavanje `history.html` u `style-for-history.css` i `script-for-history.js` datotekama. Za uređaj Klima ne dohvaćaju se stvarne vrijednosti niti se njime može zapravo upravljati, on je unesen samo radi prezentacije drugih funkcionalnosti aplikacije.

Mobilna aplikacija

U nastavku je dan prikaz mobilne aplikacije (redom: odabir sobe, popis uređaja u sobama, kontroliranje svjetla, kontroliranje roleta):



Izrađena je iOS mobilna aplikacija, što znači da je korišten Swift programski jezik. Za izradu korisničkog sučelja i navigaciju kroz aplikaciju korištena je tehnologija UIKit. UIKit pruža alate i komponente za stvaranje vizualnih elemenata i upravljanje prijelazima između zaslona. Za stvaranje view-ova (prikaza) koristio se SwiftUI, Appleov noviji okvir koji koristi deklarativnu sintaksu za izgradnju korisničkog sučelja. SwiftUI je korišten putem UIKit wrappera kako bi se iskoristile njegove moderne značajke u sklopu UIKit okvira.

Za slanje HTTP zahtjeva na server korištena je biblioteka Alamofire. Ova biblioteka omogućuje jednostavno slanje zahtjeva, upravljanje odgovorima i obradu podataka. Modele podataka spremamo u mobilnu bazu podataka Realm.

Na početnoj stranici prikazuje se popis svih dostupnih soba. Nakon odabira sobe korisniku je prikazan ekran sa popisom dostupnih uređaja u toj sobi. Za svaki uređaj prikazuje se trenutno stanje, za senzore je to očitana vrijednost, a za ostale uređaje stanje aktivnosti. Pri prvom učitavanju dohvaća se trenutna vrijednost te se nove vrijednosti dohvaćaju na zahtjev korisnika (pull to refresh). Za uređaje kojima je moguće upravljati nudi se mogućnost upravljanja, a za sve se uređaje nudi i prikaz prethodnih mjerenja. Roletama se upravlja drukčije nego ostalim uređajima, postavljen je klizni izbornik s mogućnošću spuštanja roleta od 0% do 100 % te je odabranu vrijednost potrebno predati pritiskom na Spremi gumb kako bi se akcija primijenila. Pritiskom na gumba za prikaz povijesnih mjerenja nekog uređaja, odlazi se na novu stranicu na kojoj se prikazuje zadnjih 10 mjerenja iz baze.

Komunikacija s Home Assistantom radi isto kao i kod web aplikacije, tj. koristeći klasičnu HTTP komunikaciju. Kako bi se olakšalo razvijanje korištena je biblioteka Alamofire.

Osim toga svi modeli soba i vrijednosti njihovih uređaja spremljene su u mobilnu bazu podataka Realm kako bi njihovo stanje bilo brzo dostupno i ažurno.