



TERMOSTAT

Zasnova projekta

Goran Tubić, Nikola Marin

Koper, April-Junij 2019

Uvod - definicije

Termostat je naprava, ki zazna temperaturo sistema, ter na njeni podlagi vzdržuje, zvišuje ali pa znižuje le-to.

Strežnik je fizična enota, ki preko programske opreme daje na voljo storitve drugim napravam.

Oddaljeni dostop je princip, ki omogoča uporabnikom dostop do virov s poljubne lokacije ob poljubnem času ter dovoli (ali pa ne) upravljanje virov.

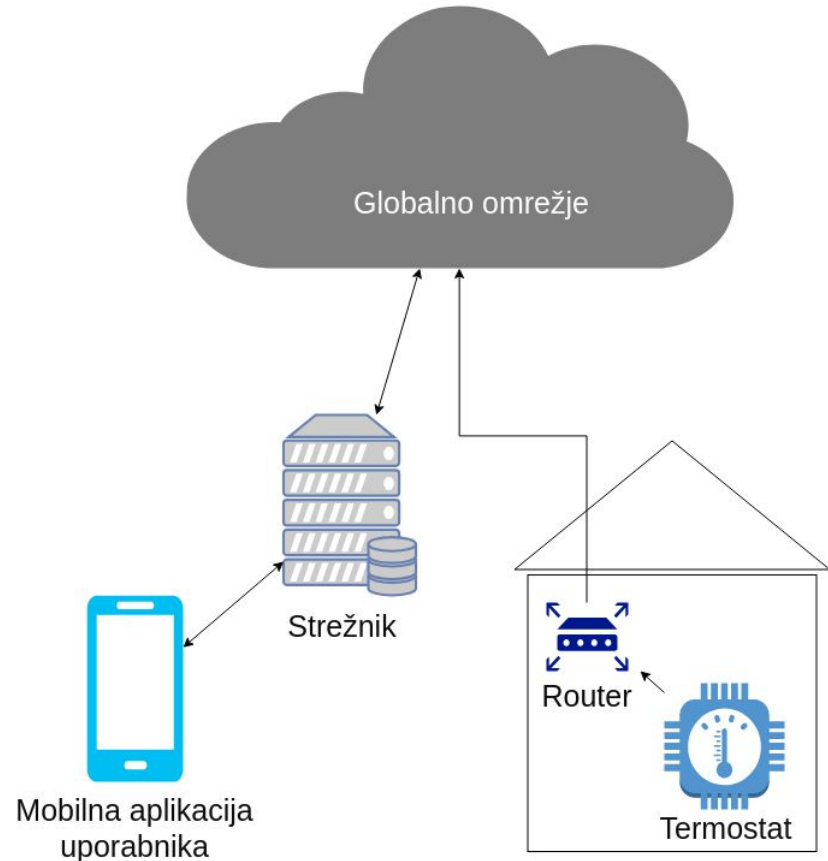
Usmerjevalnik (router) je naprava, ki povezuje dve ali več različnih omrežij.



Arhitektura sistema

Uporabnik ima pregled nad temperaturnim dogajanjem v svoji hiši. Le to lahko tudi upravlja.

Termostat je preko routerja povezan na globalno omrežje, zaradi omogočanja dostopa do lokalnega sistema.



Zahteve sistema

- V vsakem trenutku ne glede na lokacijo ali čas uporabnik hoče vedeti trenutno temperaturo
- Uporabnik hoče upravljati temperaturo ne glede na njegovo fizično lokacijo (le če ima dostop do globalnega omrežja)
- Uporabnik želi, da termostat sam uravnava temperaturo, tako, da on nima nobenih skrbi
- Uporabnik želi biti obveščen o izpadih ter “čudnih” dogajanj
- Uporabnik želi čim manjšo porabo električne energije





Zakaj še en termostat?

- uporabniki ne znajo uporabljati fizičnega termostata
- večja sledljivost in opozarjanje uporabnika v primeru izpadov
- večja želja po manjši porabi električne energije
- cena podobnih rešitev je nad 300 evrov, uporabnik pa lahko ima 5 termostatov v eni hiši
- ta termostat upošteva lokacijo uporabnika in zunanje vreme ter se prilagaja uporabniku



PLAN DELA

1. Analiza trenutnih rešitev
2. Izbira “pravilnega” fizičnega vezja za izdelavo termostata (WiFi, majhna poraba, programljivost, ...)
3. Ugotavljanje načina za postavitev dvosmerne komunikacije med termostatom iz lokalnega omrežja s strežnikom
4. Izdelava in/ali programiranje termostata, za zadostitev pogojem
5. Izdelava strežnika za komunikacijo s termostatom in priprava tega za komuniciranje z mobilno aplikacijo
6. Izdelave aplikacije
7. Testiranje delovanja sistema



PLAN DELA

1. Analiza trenutnih rešitev
2. Izbira “pravilnega” fizičnega vezja za izdelavo termostata (WiFi, majhna poraba, programljivost, ...)
3. Ugotavljanje načina za postavitev dvosmerne komunikacije med termostatom iz lokalnega omrežja s strežnikom -> [SOCKET.IO](#)
4. Izdelava in/ali programiranje termostata, za zadostitev pogojem
5. Izdelava strežnika za komunikacijo s termostatom in priprava tega za komuniciranje z mobilno aplikacijo
6. Izdelave aplikacije
7. Testiranje delovanja sistema



PLAN DELA

1. Analiza trenutnih rešitev
2. Izbira “pravilnega” fizičnega vezja za izdelavo termostata (WiFi, majhna poraba, programljivost, ...)
3. Ugotavljanje načina za postavitve dvosmerne komunikacije med termostatom iz lokalnega omrežja s strežnikom -> [SOCKET.IO](#)
4. Izdelava in/ali programiranje termostata, za zadostitev pogojem -> [ESP8266 čip](#)
5. Izdelava strežnika za komunikacijo s termostatom in priprava tega za komuniciranje z mobilno aplikacijo
6. Izdelave aplikacije
7. Testiranje delovanja sistema



PLAN DELA

1. Analiza trenutnih rešitev
2. Izbira “pravilnega” fizičnega vezja za izdelavo termostata (WiFi, majhna poraba, programljivost, ...)
3. Ugotavljanje načina za postavitev dvosmerne komunikacije med termostatom iz lokalnega omrežja s strežnikom -> [SOCKET.IO](#)
4. Izdelava in/ali programiranje termostata, za zadostitev pogojem -> [ESP8266 čip](#)
5. Izdelava strežnika za komunikacijo s termostatom in priprava tega za komuniciranje z mobilno aplikacijo -> [NODEjs](#)
6. Izdelave aplikacije
7. Testiranje delovanja sistema



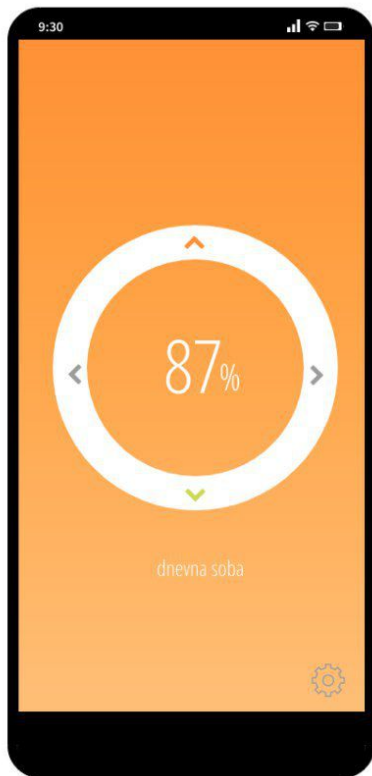
PLAN DELA

1. Analiza trenutnih rešitev
2. Izbira “pravilnega” fizičnega vezja za izdelavo termostata (WiFi, majhna poraba, programljivost, ...)
3. Ugotavljanje načina za postavitev dvosmerne komunikacije med termostatom iz lokalnega omrežja s strežnikom -> [SOCKET.IO](#)
4. Izdelava in/ali programiranje termostata, za zadostitev pogojem -> [ESP8266 čip](#)
5. Izdelava strežnika za komunikacijo s termostatom in priprava tega za komuniciranje z mobilno aplikacijo -> [NODEjs](#)
6. Izdelave aplikacije -> [IONIC Framework](#)
7. Testiranje delovanja sistema

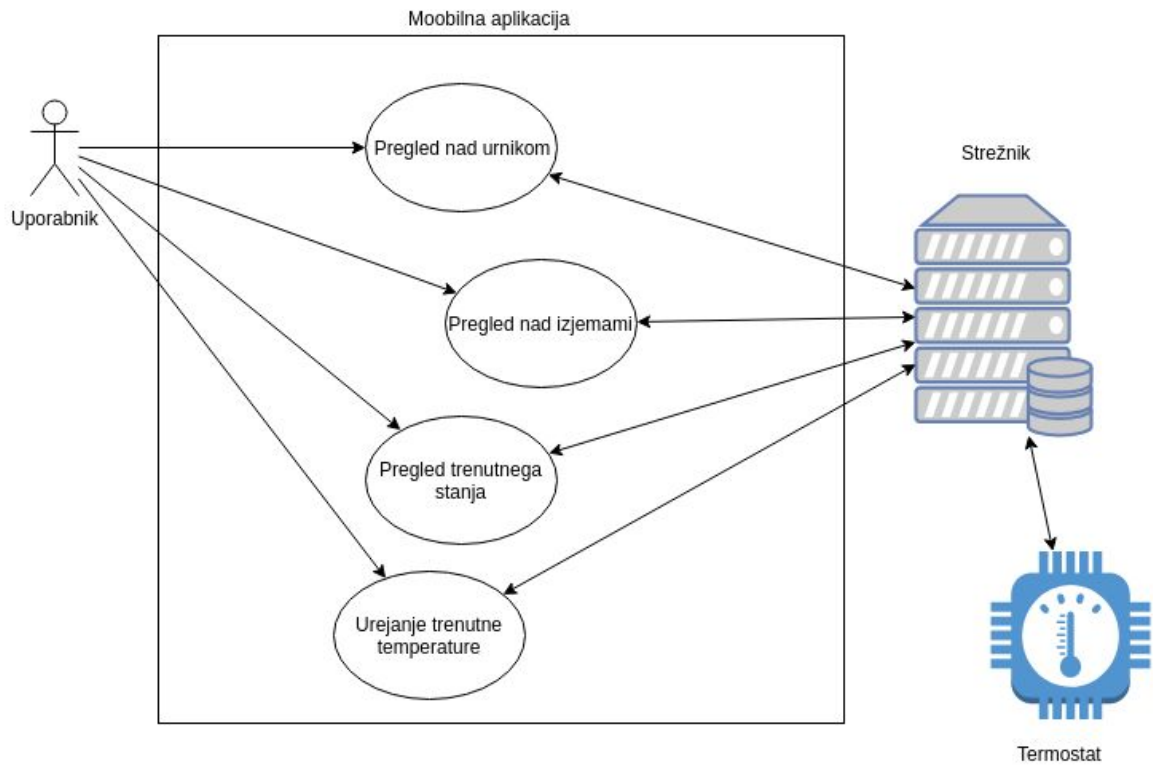


PREGLED APLIKACIJE - Uporabniški vmesnik

PREC



ik



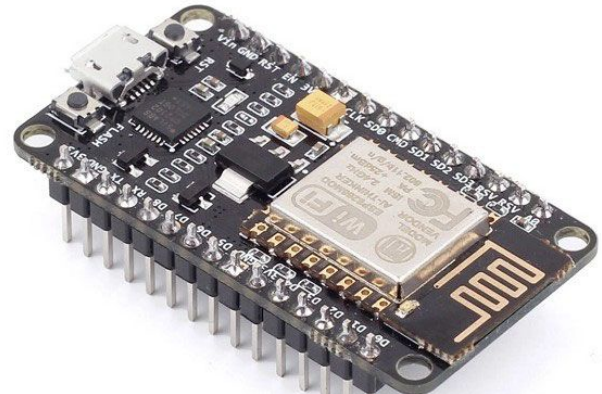


PREGLED APLIKACIJE - Uporabniški vmesnik

file:///home/gogi/Videos/ieko_v1.mp4

PREGLJED APLIKACIJE - Strojno okolje

- termostat (ESP8266 čip)
- pametni napravo (Android)
- router
- dostop do globalnega omrežja (lahko tudi brez; potem ni možno urejati na daljavo)





PREGLED APLIKACIJE - Programsko okolje

Fizično okolje	Programsko okolje
Termostat	C
Mobilna aplikacija	Ionic framework (Angular)
Strežnik	NodeJs (typescript)



PREGLED APLIKACIJE - Komunikacijsko okolje

- zelo pomembna varnost!!
- potreba po obveščanju uporabnika za izpade



PREGLJED APLIKACIJE - Komunikacijsko okolje

- zelo pomembna varnost!!
- potreba po obveščanju uporabnika za izpade

Rešitve:

- uporaba SSL certifikata za kriptiranje povezave; uporaba 512 bitnih ključev za identifikacijo termostata
- uporaba google push notifications



Možnosti nadaljnjega razvoja

V bližnji prihodnosti:

- implementacija termostata v fizičnem okolju
- dodaja urnikov in izjem

V daljni prihodnosti:

- uporaba strojnega učenja za manjšo porabo električne energije
- upoštevanje lokacije uporabnike in prilagoditev temperature glede na njegovo lokacijo
- prodaja



Hvala za pozornost!