SUUNNITELMA:

Alustava aikataulu: (ei tule pitämään 😅)

Päivä 1-2: Suunnittelu  
Päivä 3-4: Projektin alustus ja tietokantamalli  
Päivä 5: Erikoishaut ja virheenkäsittely  
Päivä 6: Testaus ja dokumentointi  
Päivä 7: Viimeistely ja raportin kirjoitus

Tavoite:  
Toteuttaa REST API joka hallinnoi tehdashallin lämpötila-antureita. API tarjoaa endpointit anturien hallintaan, lämpötilamittausten keräämiseen sekä datan hakemiseen.

Toteutuksessa käytetyt paketit

Aion käyttää projektissä samoja teknologioita/paketteja, mita kurssin aikana:

* FastAPI: Python-web-framework, joka mahdollista REST API:n helpon toteuttamisen
* SQLite: Tietokantaratkaisu
* SQLAlchemy: Mahdollistaa tietokannan käytön python-oliona ilman SQL
* Pydantic: Datan validointiin ja mallintamiseen
* Katsotaan tuleeko muuta matkan varrella

Perusteluita:

Näitä ratkaisuja käytettiin kurssin toteuttamisessa, joten ne olivat looginen valinta myös lopputyölle.

* FastiAPI on nopea ja helppokäyttöinen ja tarjoaa automaattisen laadukkaan dokumentaation
* SQLite, koska se oli vaadittuna eikä vaadi erillistä tietokantapalvelinta
* SQLAlchemy oli myös luentojen esimerkeissä, se yksinkertaistaa tietokantakyselyitä ja tekee koodista ylläpidettävämpää
* Pydantic samoin, se varmistaa, että API vastaanottaa ja palauttaa datan oikeassa muodossa

Tietorakenne:

Tietorakenne tulee todennököisesti koostumaan neljästä taulusta:

Lohkot: blocks

|  |  |  |
| --- | --- | --- |
| Nimi | Tyyppi | Kuvaus |
| id | int | Uniikki tunniste |
| name | str | Lohkon nimi (A\_13\_13 jne) |

Anturit: sensors

|  |  |  |
| --- | --- | --- |
| Nimi | Tyyppi | Kuvaus |
| id | int | Uniikki tunniste |
| sensor\_id | int | Anturin tunniste (11, 32 jne) |
| block\_id | int | Viite blocks.id |
| is\_active | bool | Onko käytössä |
| is\_error | bool | Onko virhetilassa: False |

Lämpömittaukset: temperature\_readings

|  |  |  |
| --- | --- | --- |
| Nimi | Tyyppi | Kuvaus |
| id | int | Uniikki tunniste |
| sensor\_id | int | Viite sensors.id |
| temperature | float | Lämpötila (C, 1 desimaali) |
| time | DateTime | Mittauksen ajankohta |

Tilamuutokset: sensor\_status

|  |  |  |
| --- | --- | --- |
| Nimi | Tyyppi | Kuvaus |
| id | int | Uniikki tunniste |
| sensor\_id | int | Viite sensors.id |
| is\_error | bool | Virhetila: True |
| time | DateTime | Muutoksen ajankohta |

Perustelut:

Tehtävänantona oli, että on lohkoja joiden sisällä on antureita ja antureilla on tiettyjä toimintoja. Tällöin oli loogista:

* Lohkot (blocks) ja anturit (sensors) ovat erillisissä tauluissa, koska yksi lohko voi sisältää yhden tai useamman anturin
* Lämpötilamittuakset (temp\_readings) tallennetaan erikseen, jotta niiden historiatietoja voidaan säilyttää erikseen (tämän toimivuus täytyy vielä todeta, mutta uskoisin sen ainakin teoriassa olevan näin).
* Tilanmuutoksissa (sensor\_status) sama juttu kuin lämpötilamuutoksissa: voidan seurata anturin tilahistoriaa.

Toiminnallisuudet:

Hallinnan toiminnot

|  |  |  |
| --- | --- | --- |
| Toiminto | Metodi | Endpoint |
| Lisää uusi | POST | /sensors/ |
| Muuta tilaa (normi, ei-normi) | PUT | /sensors/{sensor\_id}/status |
| Muuta lohkoa | PUT | /sensors/{sensor\_id}/block |
| Poista mittaustulos | DELETE | /sensors/{reading\_id } |

Mittaukset/hallinta:

|  |  |  |
| --- | --- | --- |
| Toiminto | Metodi | Endpoint |
| Listaa kaikki anturit | GET | /sensors/ |
| Listaa tietyn lohkon anturit | GET | /blocks /{block\_id}/sensors |
| Näytä anturin kaikki tiedot | GET | /sensors/{sensor\_id} |
| 10 uusinta mittaustulosta | GET | /sensors/{sensor\_id}/readings?limit=10 |
| Mittausten aikaväli | GET | /sensors/{sensor\_id}/readings?start=yyyy-mm-dd&end=yyyy-mm-dd |
| Anturin tilahistoria | GET | /sensors/{sensor\_id}/sensor\_status |
| Hae anturi tilan mukaan | GET | /sensors?is\_error=true |
| Virhetila graafi | GET | /sensor\_status/error-graph??????? |

Alustus ja asennukset:

Venv: python -m venv venv  
FastAPI: pip install "fastapi[standard]" (  
 - tämän jälkeen päivitetty pip: python.exe -m pip install --upgrade pip  
SQLite: pip install SQLite  
SQLAlchemy: pip install sqlalchemy

Katsotaan miten projekti etenee tästä ja miten energiat riittävät.