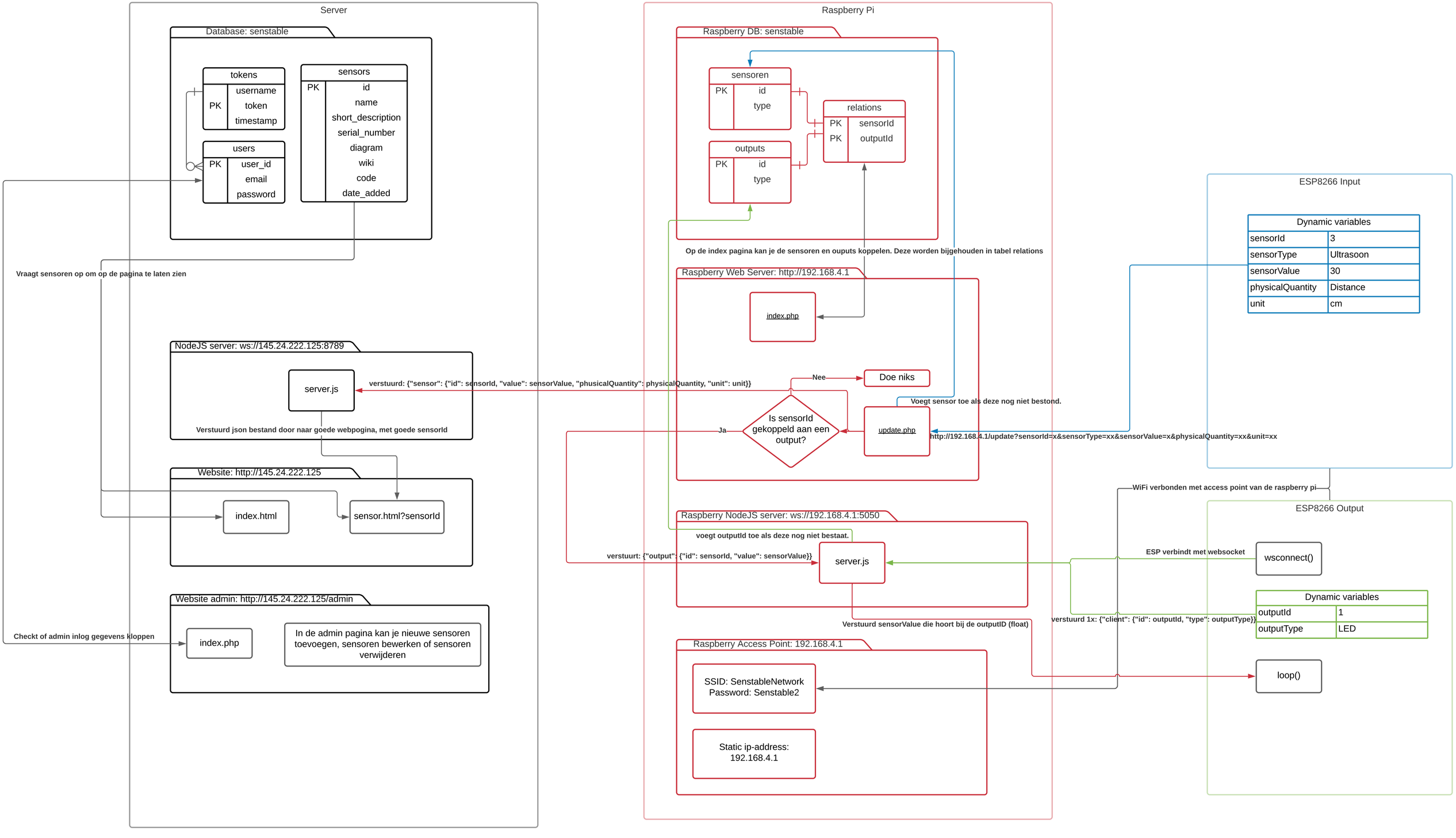
Server
Database: senstable
tokens
PK
username
token
timestamp
users
PK
user_id
email
password
sensors
PK
id
name
short_description
serial_number
diagram
wiki
code
date_added
Vraagt sensoren op om op de pagina te laten zien
NodeJS server: ws://145.24.222.125:8789
server.js
verstuurd: {"sensor": {"id": sensorId, "value": sensorValue, "phusicalQuantity": physicalQuantity, "unit": unit}}
Verstuurd json bestand door naar goede webpogina, met goede sensorId
Website: http://145.24.222.125
index.html
sensor.html?sensorId
Website admin: http://145.24.222.125/admin
Checkt of admin inlog gegevens kloppen
index.php
In de admin pagina kan je nieuwe sensoren toevoegen, sensoren bewerken of sensoren verwijderen
Raspberry Pi
Raspberry DB: senstable
sensoren
PK
id
type
relations
PK
sensorId
PK
outputId
outputs
PK
id
type
Op de index pagina kan je de sensoren en ouputs koppelen. Deze worden bijgehouden in tabel relations
Raspberry Web Server: http://192.168.4.1
index.php
Nee
Doe niks
Voegt sensor toe als deze nog niet bestond.
Is sensorId gekoppeld aan een output?
Ja
update.php
http://192.168.4.1/update?sensorId=x&sensorType=xx&sensorValue=x&physicalQuantity=xx&unit=xx
Raspberry NodeJS server: ws://192.168.4.1:5050
voegt outputId toe als deze nog niet bestaat.
verstuurt: {"output": {"id": sensorId, "value": sensorValue}}
server.js
Verstuurd sensorValue die hoort bij de outputID (float)
Raspberry Access Point: 192.168.4.1
SSID: SenstableNetwork
Password: Senstable2
Static ip-address:
192.168.4.1
ESP8266 Input
Dynamic variables
sensorId 3
sensorType Ultrasoon
sensorValue 30
physicalQuantity Distance
unit cm
WiFi verbonden met access point van de raspberry pi
ESP8266 Output
ESP verbindt met websocket
wsconnect()
Dynamic variables
outputId 1
outputType LED
verstuurd 1x: {"client": {"id": outputId, "type": outputType}}
loop()