SENSOR MONITORIN SYSTEM

Overview

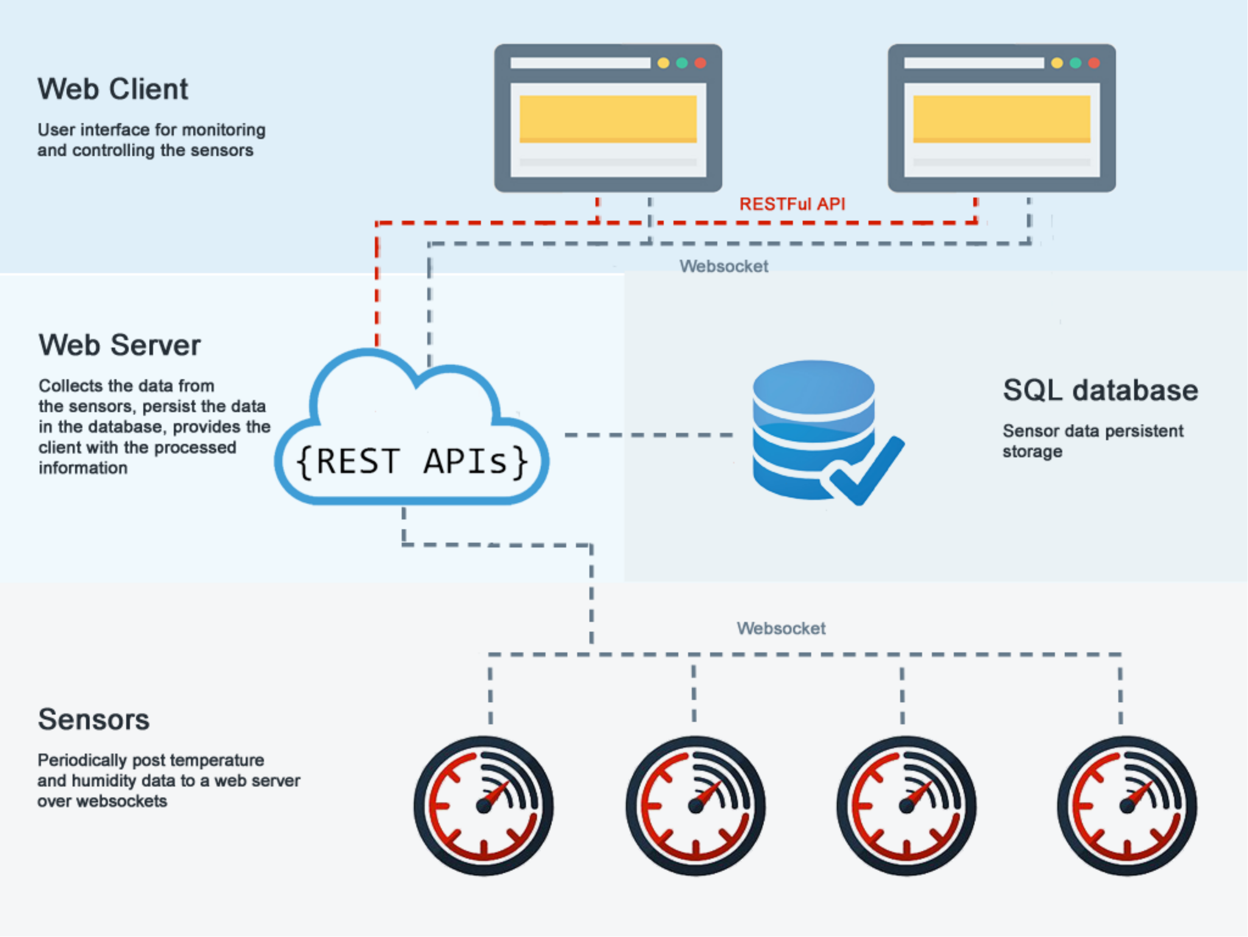
Design and implement a system, which collects and monitor data from several sensors. Provides persistence of collected data and allows user to monitor and control sensors. In order to emulate the communication, the sensors will be represented by small applications periodically posting data.

Toolset

- Front-end: ReactJS
- Back-end: NodeJS

Overview (Deliverables)

- Source code
- Database schema
- Documentation: message format between the sensors and the web server
- Documentation: how to setup and build the solution



Web Client

User interface for monitoring and controlling the sensors

Web Server

Collects the data from the sensors, persist the data in the database, provides the client with the processed information

SQL database

Sensor data persistent storage

Sensors

Periodically post temperature and humidity data to a web server over websockets

Sensor

Small applications posting periodically random data. Sensor is identified by unique number (serial number) coded into the sensor. It should send the data to the web server every 5 seconds

Web server

The web server should collect data from sensors, persist the information to database and provide exposed end-points for CRUD procedures. In order to keep it simple, no security and/or validation are needed.

API Spec:

- GET /sensors – get list of all registered sensors
- GET /sensors/{sensorId} – get the information for specific sensor
- (optional) PUT /sensors/{sensorId} – enable/disable the sensor
- Websocket /sensors/{sensorId}/data – provides real-time data for sensor measurements

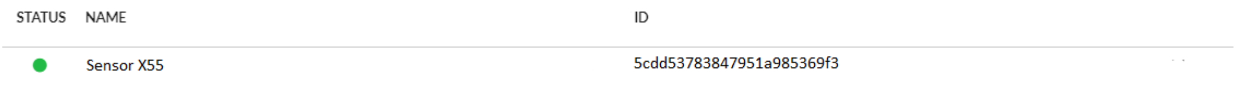
Database

Store the data collected from sensor periodically, with timestamp when data is received

Web client

The user interface for monitoring data should implement the following screens/pages:

- List of sensors with status information (Active/Inactive)



STATUS	NAME	ID
●	Sensor X55	5cdd53783847951a985369f3

- Sensor details – the design is arbitrary

Page should contain:

- Realtime measurements about temperature and humidity
- Information about sensor (id, name, etc...)
- Sensor status in realtime (active/inactive)
- Sensor measurements for last 24 hours
- (optional) Sensor control (button) to enable/disable sensor
- (optional) Graph with humidity and temperature received realtime
- (optional) Graph with humidity and temperature for last 24 hours

Auxiliary materials:

https://socket.io/, https://reactjs.org/, https://nodejs.org/en/docs/guides/getting-started-guide/, https://expressjs.com/, https://github.com/theturtle32/WebSocket-Node, https://www.sqlite.org/