IoT Shop and Device Management

Students:

Moroldo Luca

Vaccaro Fabio

Ivancich Stefano

Maino Nicola

Pham Francesco

Deronjic Denis

Supervisors:

Prof. Ferro Nicola

Dosso Denis

DEPARTMENT OF INFORMATION ENGINEERING

UNIVERSITY OF PADOVA

Project idea

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

A web platform where customer can:

- **Buy IoT devices**: wind sensor, temperature sensor, ...
- **Manage devices**: dashboard, measurements graphs, locations overview

Domain: **IoT Device Management**

Concerns:

- Provisioning & authentication
- Monitoring & diagnosing
- Configuration & maintenance

Core functionalities

Marketplace

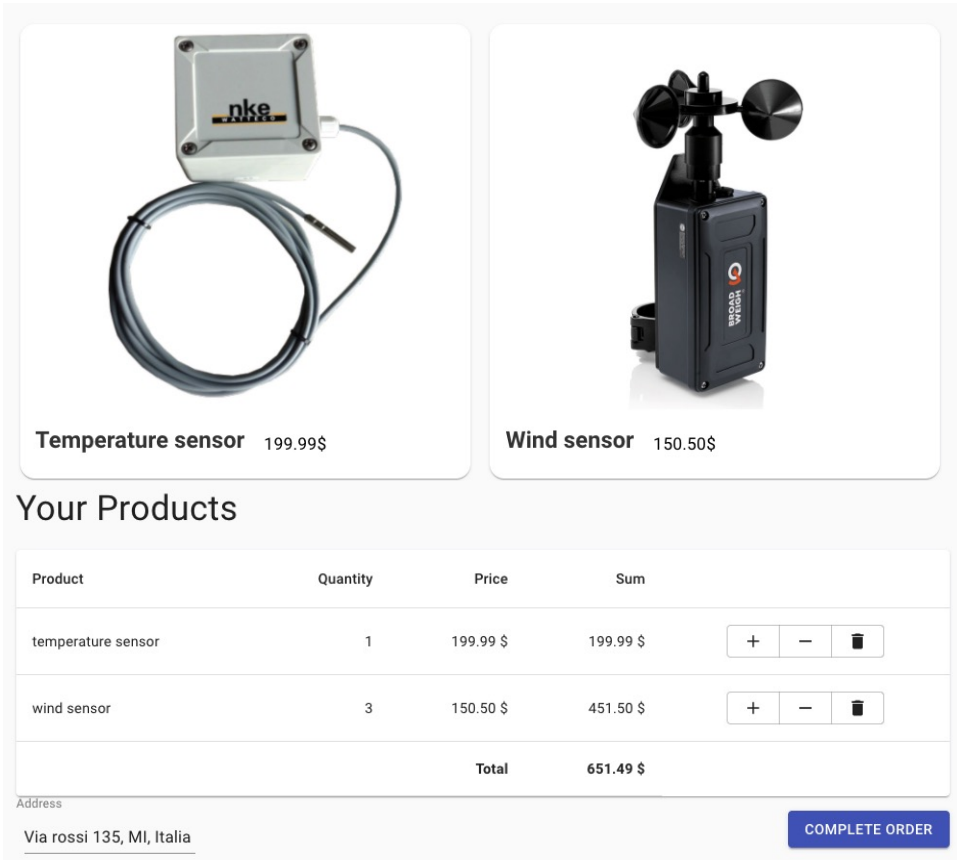
- Buy IoT devices
- Each product collects different kinds of data

Wind sensor

- Wind speed
- Wind bearing

Temperature sensor

- Temperature
- Humidity
- pressure



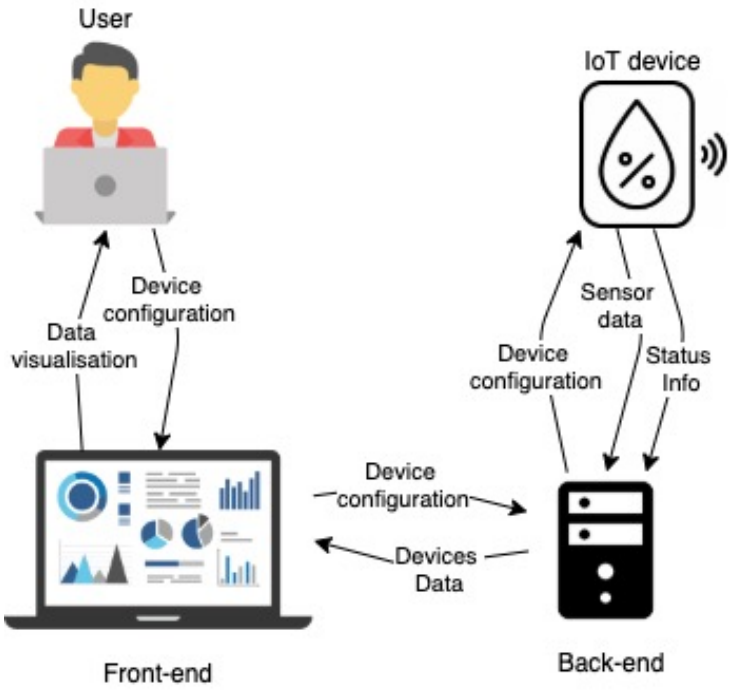
Core functionalities

Data collection

- Device authentication with bearer token
- The collected data is sent to our platform with some device status information

Device configuration

- Change device settings through the dashboard
  - Enabling/disabling
  - Publish interval
  - Location
- The device polls its configuration

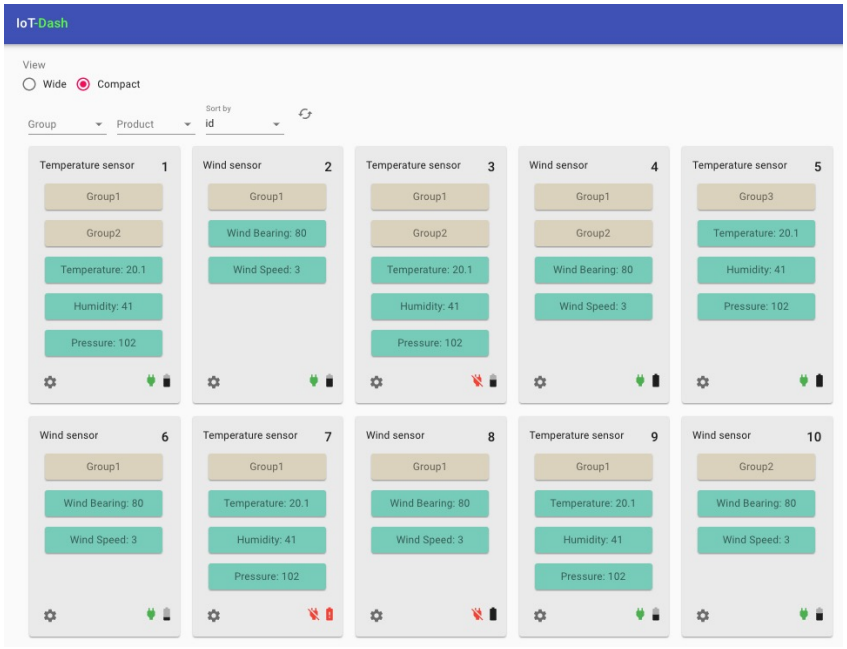


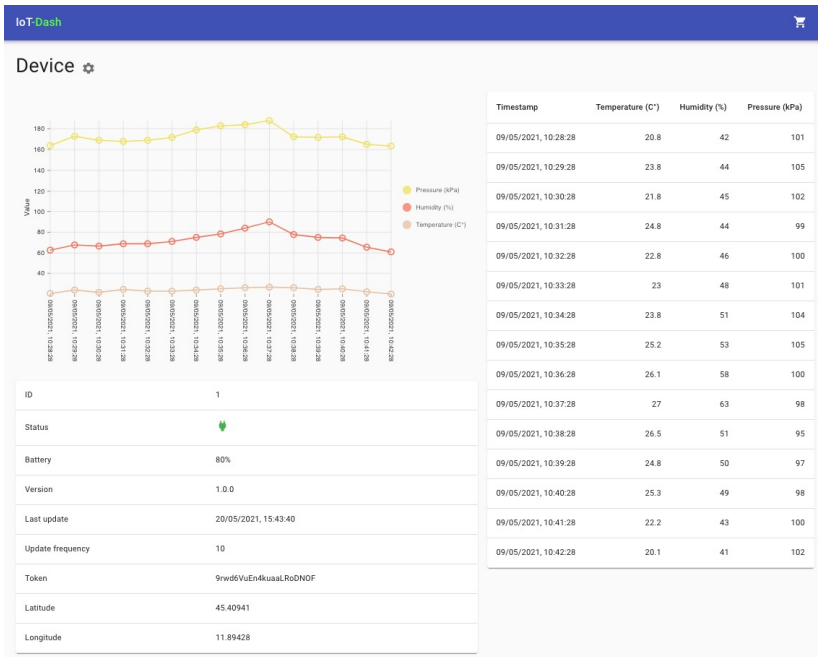
Core functionalities

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Dashboard visualization

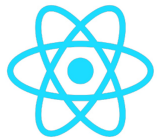
- Quick visualization of all devices
- A dedicated page showing all collected data from a device and its internal status information





Tech stack

- **Frontend**: React JS & Material UI



- **Backend**: Spring Boot (Java)

- **Database**: Postgres 13

- **Mocked devices**: Python

Project structure I

All services (e.g. frontend, backend) have been "Dockerized" and glued with Docker-Compose.

Advantages:

- **Infrastructure as Code** (IaS): code-based approach to eliminate manual processing
- **Shared development environment**: no more "but it works on my machine"

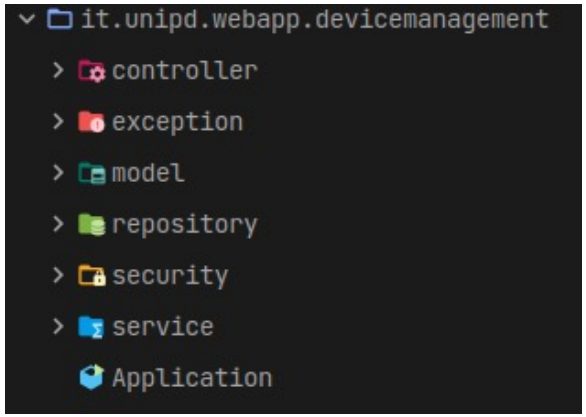
At the end, just install Docker, Docker-Compose, Make, and run:

make start

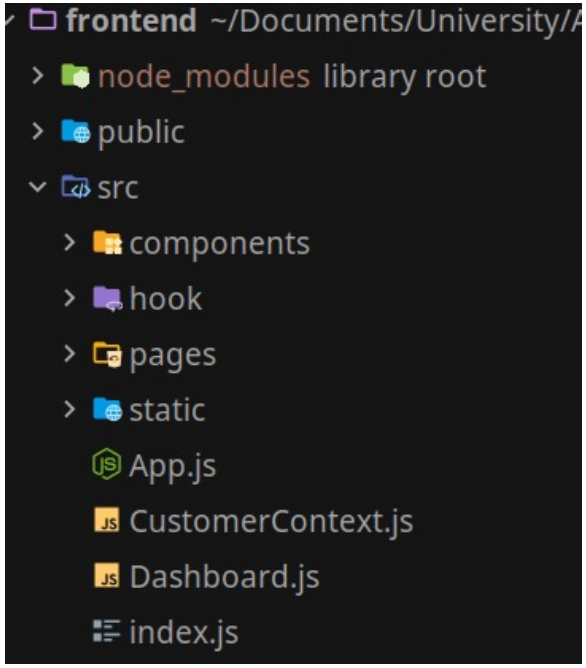
Project structure II

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Backend



Frontend



Docker Compose

```yaml
version: '3'

services:

  database-postgres:
    image: library/postgres:13.2
    container_name: database-postgres
    ports:
      - 5432:5432
    volumes:
      - ./database-postgres/data:/var/lib/postgresql/data
      - ./database-postgres/scripts:/scripts
    environment:
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
      - POSTGRES_DB=${POSTGRES_DB}

  device-management-backend:
    image: device-management-backend:latest
    container_name: device-management-backend
    ports:
      - 8080:8080
    environment:
      - SERVER_PORT=8080
      - SPRING_DATASOURCE_URL=jdbc:postgresql://database-postgres:5432/${POSTGRES_DB}
      - SPRING_DATASOURCE_USERNAME=${POSTGRES_USER}
      - SPRING_DATASOURCE_PASSWORD=${POSTGRES_PASSWORD}
      - LOGGING_LEVEL_IT_UNIPD_WEBAPP_DEVICEMANAGEMENT=${APPLICATION_LOG_LEVEL}
      - FREE_PLAN_CALLS=${FREE_PLAN_CALLS}
      - PREMIUM_PLAN_CALLS=${PREMIUM_PLAN_CALLS}

  device-management-frontend:
    container_name: frontend
    build:
      context: frontend
      dockerfile: Dockerfile
    ports:
      - 3000:3000
```

Demo

Students:

Moroldo Luca

Vaccaro Fabio

Ivancich Stefano

Maino Nicola

Pham Francesco

Deronjic Denis

Supervisors:

Prof. Ferro Nicola

Dosso Denis

DEPARTMENT OF INFORMATION ENGINEERING

UNIVERSITY OF PADOVA

Workload split

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Database Design: Collective work

UI mock-ups: Nicola – Fabio

Docker & Docker compose: Luca

Backend models: Luca

Security (device & customer auth): Luca

Generic exception handling: Luca

IoT Devices mock-up: Luca

Back-End controllers:

- Device: Francesco
- Sensor data: Fabio
- Customer: Luca
- Order: Stefano
- Group: Denis
- Product: Nicola

Front-End:

- Login/Signup/Profile: Luca
- Dashboard page: Francesco
- Device config: Nicola – Francesco
- Side bar/Top bar: Francesco
- Device page: Fabio
- Shop: Denis – Stefano
- Landing page: Denis