DCU

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

EE417 Web Application Development
Group H Project: application architecture

March 11, 2024

Chapter 1

Introduction

Unless someone else prefers to take lead, I, Michał will use this document to at least kick-start the group project. Here are my limitations:

- Fully remote, in full employment, so this, Trello, and Whatsapp are the only ways of communicating with me.
- Not a native English speaker.
- No experience as Web Developer.
- I use LaTeX to write this document.

If anyone volunteers to lead the project, I will happily follow orders.

Chapter 2

Sensor Data Streaming

The assignment requires dealing with a stream of sensor readings. As we do not have such data streams, we need to provide dummy sources of sensor readings. This will be done by implementing server-side dummy sensors that communicate with client-side (the browser).

To this end **the sensors will be registered in server-side database** and the server **will continuously provide their readings** to the clients.

```
{
    "msgkind": "reading",
    "sensor": <put sensor name here>,
    "reading": <the numeric (can be 0-1 boolean) value sent by the sensor>,
    "unit": <the unit of measured quantity, e.g. kph, decibels, or none>,
    "location": <either a string with location name or numeric coordinates>
}
```

Figure 2.1: Abstract sensor reading: JSON definition

```
{
    "msgkind": "reading",
    "sensor": "noise sensor",
    "reading": 50,
    "unit": "db",
    "location": "Room C124, Henry Grattan Building, Glasnevin Campus"
}
```

Figure 2.2: Noise sensor reading: JSON definition

```
{
    "msgkind": "reading",
    "sensor": "parking sensor",
    "reading": 1,
    "unit": null,
    "location": "Car Park 1, spot 113"
}
```

Figure 2.3: Parking spot sensor reading (busy): JSON definition

Fig. 2.1 shows the template for abstract sensor readings while Fig. 2.2 and Fig. 2.3 show how this can be specialized to two concrete cases.

2.1 Server-side events

We need to be able to:

- Register a new sensor in the database. This registration should provide all the details shown in JSON files above, plus valid reading ranges.
- Generate (every few seconds?) readings, probed from the reading ranges.
- Send the readings to the clients, using a mechanism yet to be identified.

Concerning the final point above, this: `https://stackoverflow.com/questions/5195452/websockets-vs-server-sent-events-eventsource` suggests that either WebSockets or Server-Side Events can be used.