Based on the real-life surveyed situation and the detailed system analysis in Chapter 2, Chapter 3 will center on the technologies used to develop the Data Management System. This system comprises three main components, namely, the Management Frontend, Backend, and the EPD Devices that are part of the system. The Management UI is built using NextJS and TailwindCSS to provide an easy-to-use dashboard for managing multiple devices and data efficiently. The UI also incorporates protocols to connect to HTTP, MQTT servers, and EPD devices via a Serial port. All data is processed and stored on the server using MongoDB and ExpressJS. The server communicates with EPD devices through the MQTT protocol, with RabbitMQ acting as a broker and SSL/TLS ensuring a secure connection. The EPD devices connect to the broker and subscribe to a specific topic to receive and handle any information requested by users from the front-end side, as well as send the status back to the broker.

0.1 Management UI (Front-end side)

0.1.1 NextJS

Note: because of its ..., NextJS help reduce an enourmous amount of additional third-party components, which helps creating the UI in an efficient way and also help speeding up the UI's overall responsive time.

0.1.2 TailwindCSS

because of its strength, TailwindCSS helps reducing a lot of redundant css syntax by implement it right into class, which minify JS packets when the website is rendering

0.2 Server (Back-end side)

Because of the complexity of the system that needs both processing data from client side, and also communicating with EPD devices, this Server section is divided into 2 main parts, ExpressJS server handling user's requests and storing data in MongoDB; and RabbitMQ broker communicating with devices and sending data back to ExpressJS server to process. The server used is a dedicated server hosted in Hetzner, which is a server company located in Europe, with quality is ensured to be among of the best in the world. This dedicated server also ensures the privacy, security and durable span.

0.2.1 ExpressJS Server

This part of back-end server takes reponsibilities to receive user's requests, handling to MQTT Server and store data in the system. The server is written in ExpressJS, using MVC model MongoDB is used to store all data of the system