

**Software Engineering Department**

**Braude College**

**Capstone Project Phase B**

**An Integrated Internet of Things Monitoring System – Joint work with TAMK University**

**By:**

**Dmitry Chudnovsky**

**Or Benyamin**

**Advisor:**

**Dr. Naomi Unkelos Shpigel**

**Project Code:**

**24-2-D-11**

**Project Domain:** [**https://robomo.onrender.com/**](https://robomo.onrender.com/)

**Github Phase A:** [**https://github.com/ChDimitry/FinalProjectBraude**](https://github.com/ChDimitry/FinalProjectBraude)

**Github Phase B:** [**https://github.com/RoboMo-TAMK/RoboMo**](https://github.com/RoboMo-TAMK/RoboMo)

**Website Walkthrough:** [**https://drive.google.com/file/d/170sLvRqR37K61F969uXSGDVmfJ4pZeEF/view?usp=share\_link**](https://drive.google.com/file/d/170sLvRqR37K61F969uXSGDVmfJ4pZeEF/view?usp=share_link)

**Table of Contents**

[**Abstract 4**](#_tdykid9483af)

[**1. Introduction 4**](#_qz6ctrtkweaa)

[**2. Literature Review 5**](#_7tsl95co1pe2)

[2.1 Digital Twin 5](#_6hd8njuoargy)

[2.2 Sensors and IoT 5](#_qpeyb9946hss)

[2.3 IoT and Robotics (IoRT) 5](#_meshbxprgpt0)

[2.4 Real Time Monitoring in IoT 6](#_nozp376x1642)

[2.5 IIoT 6](#_64q6qrittnbz)

[**3. Research 7**](#_w9qaks130msd)

[3.1 Scheduled Meeting 7](#_vvotpdboarmq)

[3.1.1 Meetings Summary 7](#_lz8v89t2k011)

[3.1.1.1 Meeting Number 1 - 17.6.24 7](#_k6vnupblo49w)

[3.1.1.2 Meeting Number 2 - 27.6.24 7](#_fmuiv0kvqqq0)

[3.1.1.3 Meeting Number 3 - 5.8.24 8](#_fw005z5ulcji)

[3.1.1.4 Meeting Number 4 - 11.9.24 8](#_ll4ytkl02hs3)

[3.1.1.5 Meeting Number 5 - 6.11.24 9](#_wrvhakjq9gdw)

[3.1.1.6 Meeting Number 6 - 12.12.24 9](#_5gwqc8skryfj)

[3.1.1.7 Meeting Number 7 - 9.1.25 9](#_q19dxbgqxgi8)

[**4. Engineering Process 10**](#_5hs3qa424ug3)

[4.1 Development Process 10](#_q58isaungeh6)

[4.2 User-Centered Design Approach 10](#_9ookozug85u9)

[4.2.1 Regular Customer Meetings 10](#_idb7hppp9r1w)

[4.2.2 Prototype-Driven 10](#_22mk7gvvfly9)

[4.2.3 Conclusion 10](#_xarypomr9ajr)

[4.3 Workflow 11](#_mwzye4trtxzj)

[4.4 Architecture Diagram 11](#_d2g6o03eqvd)

[4.5 Technologies Review 12](#_ylqyib19jb67)

[4.5.1 Websocket 12](#_g8pne0peef39)

[4.5.2 Client-Side Technologies 12](#_al197spsem1n)

[4.5.2.1 React 12](#_bt40ny6ay93h)

[4.5.2.2 Javascript 12](#_rwssqvo0g2i0)

[4.5.2.3 Tailwind Css 12](#_rv21uara9s14)

[4.5.3 Server Technologies 12](#_xcxjw2qayyuj)

[4.5.3.1 Node.js 12](#_wxorirko9od6)

[4.5.3.2 Fiware 12](#_wknc8h9dxg9a)

[4.5.3.3 Firebase 13](#_m5fbff7v9p93)

[**5. Work Artifacts 13**](#_7pnf0ki13gi0)

[5.1 Requirement 13](#_3q460vq8uiz6)

[5.1.1 Functional Requirements 13](#_niziijl1zcsu)

[5.1.2 Non-Functional Requirements 14](#_euodehormc1f)

[5.2 Use Case 15](#_nwwf4loqye1b)

[5.3 Activity Diagram 16](#_vorzpplhanp8)

[**6. Evaluation 17**](#_n3bvzxe0ubke)

[6.1 Feedback Analysis 17](#_qyuc80f0hcuz)

[6.1.1 Positive feedback 17](#_wjbjcxcto7tu)

[6.1.2 Areas for Improvements 17](#_1wzn8eumrzew)

[6.2 SUS Survey summary 18](#_obu3xsx76681)

[**7. Conclusion and Summary 18**](#_v5jjsadvygb2)

[7.1 Achievements 19](#_sgs48m9apx72)

[7.2 Challenges and Solutions 19](#_nj0yi2d27glm)

[**8. Testing Plan 20**](#_q8inq6bxf91c)

[8.1 Scope 20](#_6919m7xaqjxk)

[8.2 Objectives 20](#_i4dj70mwtevm)

[8.3 Testing Approach 20](#_89ehe84fpt0z)

[8.4 Test Cases 21](#_i0ai9qf2bzp1)

[**9. AI Tools and Prompts 23**](#_9xybziwumhwn)

[9.1 Prompts Used 23](#_1rbvt9mf4ol1)

[**10. User Guide 23**](#_npu1jewjw2bq)

[10.1 Login Page 23](#_7pqijw1lj3oh)

[10.2 Sign Up Page 24](#_g5uoy3wow5qj)

[10.3 Homepage 25](#_tbnw43gkfcxa)

[10.4 Widget bar 26](#_59v0nn9n97yw)

[10.5 Devices list 26](#_acdb8bvm2q4f)

[10.6 Configuration section 28](#_kelky4ox9exk)

[10.7 Map 28](#_jmqfo4u2amah)

[10.8 Device attribute 30](#_jvvs67uwubpk)

[10.9 Compare section 31](#_tgrj4hfplevb)

[**11. Maintenance Guide 34**](#_uqqfxt2bad06)

[11.1 Installation 34](#_gzlxon89jfwc)

[11.1.1 Prerequisites 34](#_jam4qnf2veat)

[11.1.2 Setup 34](#_ukn9ors8c0vs)

[11.2 Deploying the website 34](#_5lmunpk6otho)

[11.2.1 Client side 34](#_z3ngarmyohcf)

[11.2.2 Server side 35](#_p4f11jz79c31)

[11.3 Database 35](#_gt3ypxab0hhq)

[11.3.1 Authentication 36](#_akqpkdyb03i0)

[11.3.2 Firestore Database 36](#_ecq93xy1by0h)

[11.4 Server requests 37](#_mdj3x9av4qqs)

[**12. References 37**](#_8bheexdr71t2)

[**13. Appendices 39**](#_311you43qf1n)

[13.1 Meetings Capture 39](#_2n2c03pmalj)

### 

### Abstract

In recent years, the Industrial Internet of Things (IIoT) has emerged as a transformative force in modern industrial processes, revolutionizing data collection, analysis, and utilization across manufacturing and research environments. This project addresses a specific challenge faced by TAMK, which operates a laboratory equipped with autonomous robotics featuring various types of sensors. The main challenge of the laboratory lies in the absence of a comprehensive system to effectively monitor and transmit live data from these diverse robotic units.

In this inter-institutional collaboration, TAMK provided an API that contains the collected data from their robotics laboratory sensors. Our contribution focused on developing a web application that connects to this API to access, process and visualize the sensor data. This solution creates a unified platform for data collection, monitoring, and analysis. The system displays and monitors real-time data collected from the sensors on the robotics, providing users with the ability to visualize and analyze sensor data, track changes over time, and gain insights into the system's performance and conditions.

The primary contribution of this system lies in its capacity for remote communication and display, serving as a proof of concept for an inter-institutional IIoT framework. This project emphasizes cooperation between TAMK and Braude, highlighting the potential for collaborative efforts in the realm of Industry 4.0. Through regular communication and agile methodologies, we delivered a robust solution that marks a significant milestone as Braude's first Industry 4.0 initiative. The collaboration between the two institutions demonstrates the potential of inter-institutional partnerships in driving innovation in IIoT and setting a foundation for future developments in the field.

### 1. Introduction

Tampere University of Applied Sciences (TAMK) in Finland stands at the forefront of innovation in applied sciences and technology. At the heart of its cutting-edge facilities lies a state-of-the-art robotics laboratory, serving as a microcosm of the future smart factory. This advanced facility houses an impressive array of autonomous robotic units, each equipped with various sensors capable of collecting real-time data on different aspects of their operation and environment. The laboratory provides an ideal setting for students, researchers, and industry partners to explore the integration of robotics, sensors, and data analytics in conditions that closely mimic actual manufacturing environments. Despite its advanced equipment, TAMK's robotics laboratory faced a significant challenge: the absence of a comprehensive system to effectively monitor and transmit live data from its diverse robotic units. This limitation hindered the full realization of the laboratory's potential as a testing ground for Industrial Internet of Things (IIoT) applications. To address this, TAMK initiated a collaborative project with Braude College of Engineering in Israel, aiming to develop an integrated IIoT software system. This book documents the journey of developing this system, exploring the challenges faced, the solutions devised, and the lessons learned throughout the process. By bridging the gap between academic research and practical application, this project not only enhances the capabilities of TAMK's robotics laboratory but also serves as a proof of concept for inter-institutional IIoT frameworks in the realm of smart manufacturing and research facilities.

### 2. Literature Review

#### 2.1 Digital Twin

Digital Twin technology represents a transformative advancement with potential applications spanning manufacturing, healthcare, and smart cities. There are evidences that emphasize its promise in enhancing performance and enabling predictive maintenance.

On one hand, we can understand that data integration and security are crucial for realizing the full benefits of Digital Twins [2]. On the other hand, there is a necessity for a standardized definition to alleviate confusion and promote consistent understanding across industries [10].

Together, these insights stress the importance of overcoming technological and definitional hurdles to fully leverage the capabilities of Digital Twin technology.

#### 2.2 Sensors and IoT

The integration of IoT and advanced sensor technologies is revolutionizing robotics in agriculture and industry.

Autonomous farming robots, equipped with various sensors and image processing capabilities, use IoT and wireless sensor networks to detect weeds, monitor environmental parameters, and navigate fields, enhancing precision and efficiency while reducing costs [5]. Effective decision-making in IoT applications relies on data processing methods like denoising, outlier detection, and the integration of cloud, fog, and edge computing [6].

In industrial robotics, sensors such as tactile, visual, laser, and inertial are vital for tasks like human-robot collaboration and navigation [7]. Advancements in these sensors and data processing algorithms are making robots more intelligent and independent, despite the challenges of meeting increasing performance demands.

The integration of IoT and advanced sensors in agriculture and industry is enhancing robotic precision, efficiency, and intelligence while reducing costs, though challenges in performance optimization remain.

#### 2.3 IoT and Robotics (IoRT)

The integration of the Internet of Things (IoT) with robotics represents a rapidly evolving area with significant technological and practical implications.

There are two which explore the integration of IoT with robotics, highlighting its technological implications and real-world applications. The first, delves into the feasibility and challenges of IoT-aided robotics, noting issues with protocols, communication technology, and security, and emphasizing the need for further research despite the technology's maturity [3] . The second, provides a comprehensive overview of existing concepts and architectures, underscoring the benefits and challenges of IoT in robotic systems [4].

Additionally, integrating robotics with IoT was explored to form the Internet of Robotic Things (IoRT) [9]. It provides an understanding of IoRT's principles, proposes new frameworks, and highlights key research challenges. In a conducted research, the authors conclude that while IoRT holds significant potential, it presents unique technical challenges that need to be addressed [11]. By combining sensing, AI, and IoT, IoRT aims to create smart, autonomous systems, significantly advancing IoT by integrating AI, robotics, and machine learning [ibid]. These insights illustrate that while IoT and IoRT-enhanced robotics offer significant potential, addressing technological and research gaps is essential for advancing the field.

Together, these insights illustrate that while IoT-enhanced robotics offers significant potential, addressing technological and research gaps is essential for advancing the field.

#### 2.4 Real Time Monitoring in IoT

Real time monitoring and control systems are vital in environmental and industrial applications due to their ability to provide immediate data and feedback, enabling quick decision making and response.

One example of a robot for real time environmental and industrial monitoring is the “InterBot 1.0” [8]. The purpose of such a robot is to leverage advanced sensor technology for measuring temperature, humidity and gas levels. It features real time data visualization and allows for continuous monitoring and analysis.

The ability of providing immediate data and feedback enables rapid responses to emerging issues, supports predictive maintenance, and enhances overall system efficiency and reliability.

#### 2.5 IIoT

The Industrial Internet of Things (IIoT) refers to the network of interconnected sensors, instruments, and devices used in industrial applications such as manufacturing, energy management, and logistics. These devices collect and analyze data in real time to enhance operational efficiency, productivity, and decision-making.

There are some advantages for IIoT which include enhanced efficiency and productivity, improved data collection and analysis, cost saving, remote monitoring and control [1]. However, the implementation of IIoT is not without challenges, particularly concerning security and privacy. IIoT systems are susceptible to cyberattacks and hacking, compatibility issues, lack of standardization, and high implementation costs, all of which pose significant barriers to widespread adoption [ibid]. Despite these disadvantages, the potential of IIoT to drive innovation and maintain a competitive edge in the industrial sector is immense.

In conclusion, while the Industrial Internet of Things (IIoT) provides significant benefits like enhanced efficiency, productivity, and cost savings, it also faces challenges in security and privacy. Addressing these issues through robust privacy measures and standardized protocols is essential for industries to fully realize IIoT's potential and maintain a competitive edge in the technological landscape.

### 3. Research

#### 3.1 Scheduled Meeting

During our work in the semester we had scheduled meetings with Senior Lecturer Mr. Kari Naakka on behalf of TAMK University. Holding meetings with TAMK University on a regular basis constituted an important and main focus during our work. During these meetings we introduced the work that was done and the progress we had in that period of time. Moreover, these meetings contributed to our understanding of topics and points that were not clear to us. At the end of every meeting we agreed on a date for the following meeting and the progress we are going to have for that meeting.

#### 3.1.1 Meetings Summary

#### 3.1.1.1 Meeting Number 1 - 17.6.24

In this meeting we held the first update on the process being done until that point. We introduced a first look on the website, a basic website without map included. We talked about the device's positions on the site and how to transfer this using the API. Furthermore, we solved the issue we had regarding connecting to TAMK’s server using the OpenVPN application. We asked TAMK’s representative about a single API which gathers together all the devices and agreed to get an update until the next meeting.

#### 3.1.1.2 Meeting Number 2 - 27.6.24

In this meeting we held an update on what has been done so far. We have shown a preliminary prototype of the site as for that point, which includes 3 devices on a map and when you click on a device, a menu opens where you can see the name of the devices and its features. When clicking on an attribute, you can place it at the top of the list, copy its content and create a visualled graph for it. During the meeting few points were suggested by the customer to think ahead:

* Scaling: choose a time frame.
* Real time meter: bar or needle for the data being displayed. As a result we added a widget which indicates about the amount of data that is transferred from the server.
* Showing Automatic update of data by the website. We decided to refresh the displayed information at a constant interval. This happens by pulling in new data every fixed time interval.
* Settings file: file which contains the beginning location, floor plan for more dynamic interface.

The customer mentioned a point to look ahead when the website is complete, having a 3D model. This depends on getting the model that needs to be presented.

We ended the meeting by scheduling the next meeting to the 5th of August where we will show the progress made in the Phase A Book project.

#### 3.1.1.3 Meeting Number 3 - 5.8.24

In this meeting we first showed the customer the progress we had on the book project. We showed him the sections we have done until the meeting and asked him for feedback. The customer was very excited to see the progress we have done. As a result, he asked for few things:

* The customer asked us to send him a copy of the book we have done so far. We agreed to send him a copy of the final book once we finish working on that.
* The customer told us that the name of the system is up to us and we can decide how to call the system. We decided to call the system “RoboMo” which stands for “Robotics Monitoring”.
* The customer asked us to deploy the current website and share it with a link for him to show the progress to other members in the TAMK university and could have a feeling on the website. We agreed to start working on deploying once we finish working on the website’s structure.

At the end of the meeting we agreed on a date for the following meeting.

#### 3.1.1.4 Meeting Number 4 - 11.9.24

In this meeting we first talked with the customer about the comments he had on the copy of the book. The customer emphasized the importance of mentioning him with the words “The customer” and not his private name except for one time when introducing him. In addition, he suggested thinking about different names for the project that will replace the words “software system” with something more specific. The customer also suggested changing the way the requirements are presented from “high level requirements” to “low level requirements”.

Then we started talking about the website itself and the changes we had from the last meeting. We explained that changing the structure of the website required us to start from the beginning and therefore there are some changes from the last meeting. We asked the customer about the 3D model that required to be shown on the website and he informed us that the system responsible for taking pictures of the devices is still not available and will be ready only at the end of the year (see picture 1 in section 13.1).

We agreed on few points to be updated:

* The customer will update us in the next 2 weeks when the data of the robot’s position will be ready.
* Considering the situations where a robot moves fast and slow, and the way it will represent on the map in the website.
* Thinking of a way of comparing 2 sensors' data in the same page. At this point the page will display only the data and will create a graph of a specific device. The customer suggested improving this by enabling it for different devices at the same time and option for different scales for each graph. As a result we decided to have the option of creating a graph without the need of first selecting the device and device’s attributes. In that way the comparing window is independent and different devices and different device’s attributes can be used for creating graphs.
* The customer will update us when the data on the 3D model will be available.

At the end of the meeting we agreed on November 6th as our next meeting.

#### 3.1.1.5 Meeting Number 5 - 6.11.24

In this meeting, we presented the customer with the changes made after the previous meeting. We showed him the new way of creating graphs which enables the option to compare between different types of attributes of the same device and between attributes of different devices (see picture 2 in section 13.1).

At the end of the meeting we agreed on December 12th as our next meeting, where we will provide feedback about the deployment of the website.

#### 3.1.1.6 Meeting Number 6 - 12.12.24

In this meeting, we talked with the customer about the difficulties that arise when trying to deploy the website. We showed the customer that we succeeded in deploying and running the client side of our project using Render but when trying to deploy the server side we faced problems regarding the OpenVPN. We asked the customer about the importance of having the OpenVPN on when trying to access the API and we asked if there is any possible way of accessing the API without using the OpenVPN. The customer informed us that there is no option of not using the OpenVPN when trying to access the API because it's mandatory.

Moreover, we showed the customer the evaluation questionnaire we made and asked for feedback. The customer asked us to send him a link to the questionnaire so he will be able to review it.

At the end of the meeting we agreed on January 9th as our next meeting, where we planned to succeed in deploying the server side and show it to the customer.

* After the meeting the customer replied to us in email after reviewing the evaluation questionnaire and suggested adding a section in the questionnaire where users would fill in suggestions for improvement.

#### 3.1.1.7 Meeting Number 7 - 9.1.25

In this meeting, we showed the customer the deployed website. While presenting the website, the customer asked his student to activate one of the robots in the laboratory to check if the data is changed live. As a result, we could see the data changing live.

At the end of the meeting we agreed to continue working on the website and make it ready for evaluation by students in TAMK. The customer asked us to add some introduction at the beginning of the questionnaire (see picture 3 in section 13.1).

* After the meeting, we added the introduction at the beginning of the questionnaire. We also added 2 tasks in the introduction, for the students filling the questionnaire to do in the website before filling the questionnaire.
* We updated the customer by email about the changes made for his comments on it.

### 4. Engineering Process

#### 4.1 Development Process

In this section, we will describe our workflow, the methodologies and technologies used in order to build a system which is capable of monitoring and analyzing data collected from various types of robots. This includes data processing, backend frameworks for handling data streams, and frontend technologies for creating dynamic, real-time visualizations. Our approach balanced the need for robust, scalable data handling with flexible, intuitive user interface, ensuring a final product that could adapt to different robot types while remaining powerful and user-friendly.

#### 4.2 User-Centered Design Approach

Our project employs a user-centered design (UCD) approach, placing the customer's needs and requirements at the forefront of our development process. This methodology ensures that the final product not only meets technical specifications but also aligns closely with user expectations and usability requirements.

#### 4.2.1 Regular Customer Meetings

As detailed in our meeting summaries (sections 3.1.1.1 to 3.1.1.7), we conduct regular meetings with the customer. These meetings serve as crucial touchpoints for:

1. Presenting progress and gathering immediate feedback
2. Clarifying requirements and expectations
3. Discussing challenges and potential solutions
4. Planning next steps and setting priorities

#### 4.2.2 Prototype-Driven

We developed and presented prototypes to the customer at various stages:

1. Initial prototype presentation (Meeting 1 - 17.6.24)
2. Updated prototype with customer-requested features (Meeting 2 - 27.6.24)

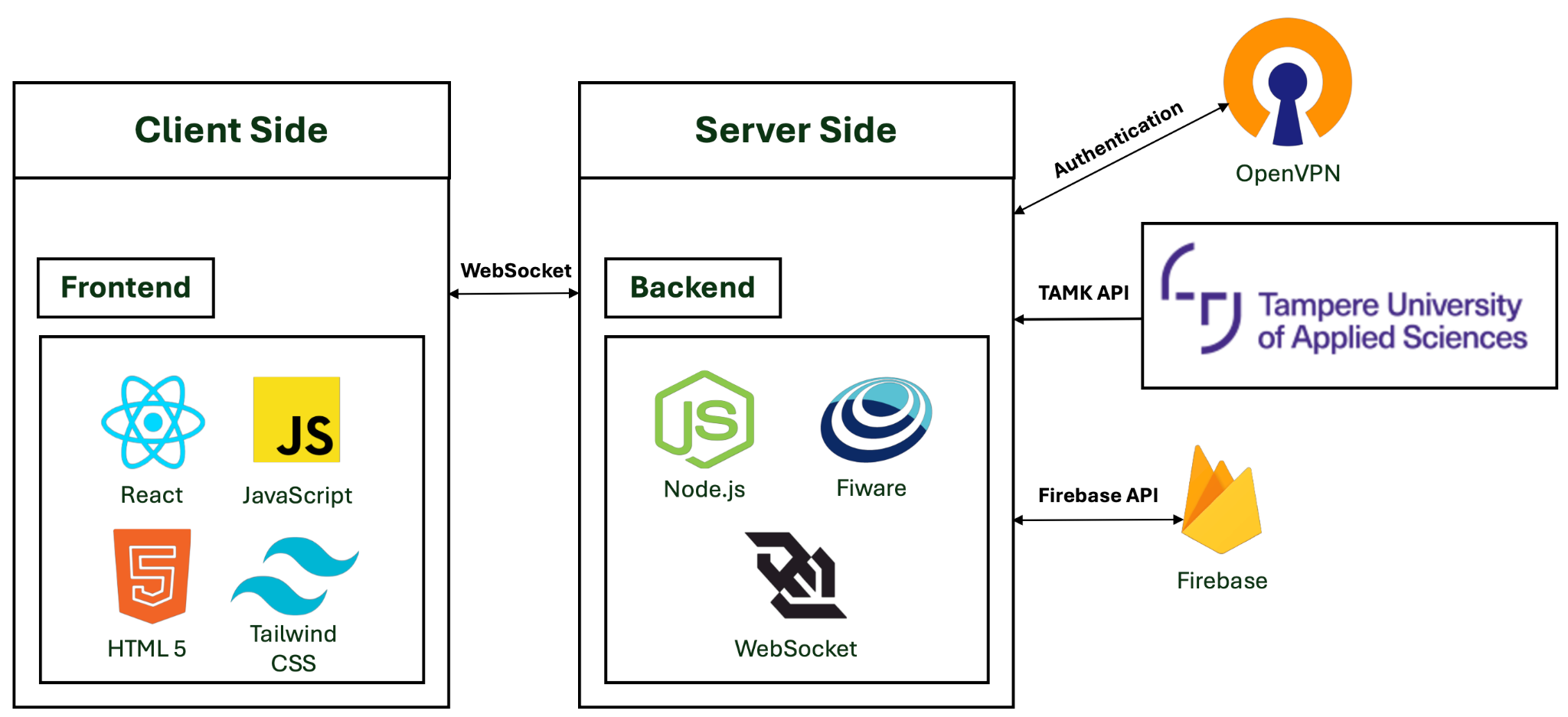
#### 4.2.3 Conclusion

By adopting a user-centered design approach, we ensure that our solution not only meets technical specifications but also provides a seamless, intuitive experience for the customer. This approach was instrumental in shaping our development process and continued to guide our efforts throughout the project development process.

#### 4.3 Workflow

*figure 1. Workflow diagram*

#### 4.4 Architecture Diagram

****

*figure 2. Architecture diagram*

The diagram (figure 2) illustrates our architecture which is combined by two main components: client side and server side. The client side is built on 4 main parts which are: react, tailwind, html and javascript. The server side is built on 2 main parts which are the nodejs and Fiware. We are using Firebase for user management and we are using OpenVPN in order to get access to the API that stores the collected data from the robotics and sensors in the laboratory.

#### 4.5 Technologies Review

#### 4.5.1 Websocket

WebSocket is a bidirectional communication protocol that can send the data from the client to the server or from the server to the client by reusing the established connection channel. The data is continuously pushed/transmitted into the same connection which is already open [22]. Unlike HTTP, where the client has to keep requesting updates (pooling), WebSockets allow the server to push updates to the client as soon as data changes.

#### 4.5.2 Client-Side Technologies

#### 4.5.2.1 React

React is a free and open-source front-end JavaScript library [12]. React is a library for helping developers build user interfaces as a tree of small pieces called components. This component is a mixture of HTML and JavaScript that capture all of the logic required to display a small section of a larger UI [13].

#### 4.5.2.2 Javascript

JavaScript is a lightweight programming language commonly used by web developers to add dynamic interactions to web pages, applications, servers and even games. JavaScript’s widespread applications in web, mobile app and game development make it a valuable language to learn [14].

#### 4.5.2.3 Tailwind Css

Tailwind css is a low-level framework. Unlike other css frameworks like Bootstrap, Tailwind doesn’t offer fully styled components like buttons, dropdowns and navbars. Instead, it offers a utility class which gives the option to create our own reusable components. Tailwind css provides a lot more flexibility and control over what the application looks like than other CSS frameworks [15].

#### 4.5.3 Server Technologies

#### 4.5.3.1 Node.js

Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client’s browser. Developers use Node.js to create server-side web applications and it is perfect for data-intensive applications since it uses an asynchronous, event-driven model. Node.js operates on a single-threaded event driven architecture, utilizing an event loop to handle multiple concurrent operations without blocking. Node.js’s non blocking, event-driven architecture makes it ideal for real-time applications, web servers, APIs and more [16].

#### 4.5.3.2 Fiware

FIWARE is an open source initiative that works towards building a set of standards to develop smart applications for different domains such as Smart Cities, Smart Ports, Smart Logistics, Smart Factories and others. FIWARE promotes a standard that describes how to collect, manage and publish context information, and additionally adds certain elements that allow exploiting collected data [17].

The FIWARE platform provides a rather simple yet powerful set of APIs that ease the development of smart applications in multiple vertical sectors [18].

#### 4.5.3.3 Firebase

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. The Realtime Database is a NoSQL database and as such has different optimizations and capabilities compared to a relational database [19].

### 5. Work Artifacts

#### 5.1 Requirement

#### 5.1.1 Functional Requirements

| **No.** | **Requirement** |
| --- | --- |
| 1 | The system will allow the administrator to approve new sign-ups |
| 2 | The system will allow users to select devices from a map |
| 3 | The system will allow users to choose which devices are shown on the map |
| 4 | The system will allow users to choose attributes for each device |
| 5 | The system will allow users to copy selected data |
| 6 | The system will allow users to mark an attribute as a favorite |
| 7 | The system will allow users to hide a device's attributes |
| 8 | The system will allow users to expand the information window |
| 9 | The system will allow users to minimize the information window |
| 10 | The system will allow users to add graphs of an attribute to the compare window |
| 11 | The system will allow users to remove graphs from the compare window |
| 12 | The system will allow users to view the device in 360 degrees |
| 13 | The system will allow users to close the selected device |
| 14 | The system will allow access to the site through mobile devices |
| 15 | The system will support dark mode functionality |
| 16 | The system will allow users to log in |
| 17 | The system will allow users to sign up |

*table 1. Functional Requirements*

#### 5.1.2 Non-Functional Requirements

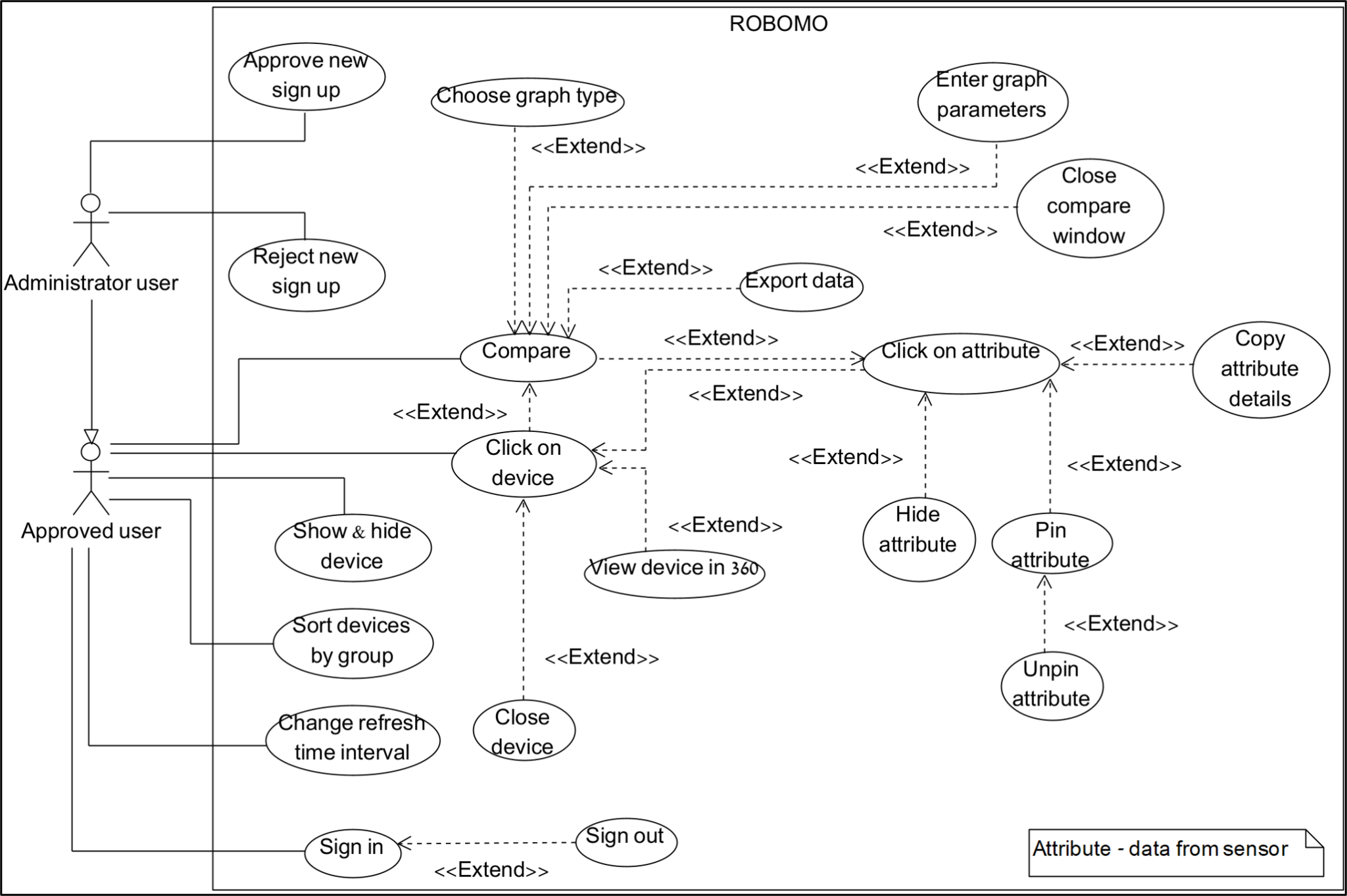
| **No.** | **Requirement** | **type** |
| --- | --- | --- |
| 1 | The site should be easy to use and quick - 85% of users should be able to complete key tasks without assistance | Usability |
| 2 | The site should be easy to maintenance and update - Code modularity should support updates with minimal downtime (<5 minutes) | Maintainability |
| 3 | The site should be mobile friendly (WCAG 2.2, Web Content Accessibility Guidelines 2.2 - 2023) [21] | Adaptability |
| 4 | The site should be up to date | Maintainability |
| 5 | The site should be lightweight (e.g.,load time up to 3 seconds) | Efficiency, Performance |
| 6 | The site will save cookies | Usability |
| 7 | The site will work properly even if non of the devices are available | Fault tolerance |
| 8 | The system should support up to 5000 numbers of users connected at the same time without degradation in performance | Scalability |
| 9 | Only authorized users should be able to access the data - Multi-factor authentication (MFA) for all access points | Security |
| 10 | The system should process and display search result, filtering and sorting operations within 1 seconds | Performance |
| 11 | The system should comply with relevant data protection regulations (e.g., GDPR) | Compliance |
| 12 | The system should be designed to accommodate future enhancements with minimal changes - New modules or features can be integrated with no more than 10% modification of existing code | Extensibility |
| 13 | The system should provide interactive features that allow users to manipulate and visualize data in real-time - Response time for interactive queries should be < 1 second | Interactivity |
| 14 | The system’s components should be modular and reusable in other projects - At least 80% of system components should be reusable in similar projects | Modularity |
| 15 | The system should complete page refresh operations within 1.5 seconds under normal load conditions | Performance |

*table 2. Non-Functional Requirements*

#### 5.2 Use Case

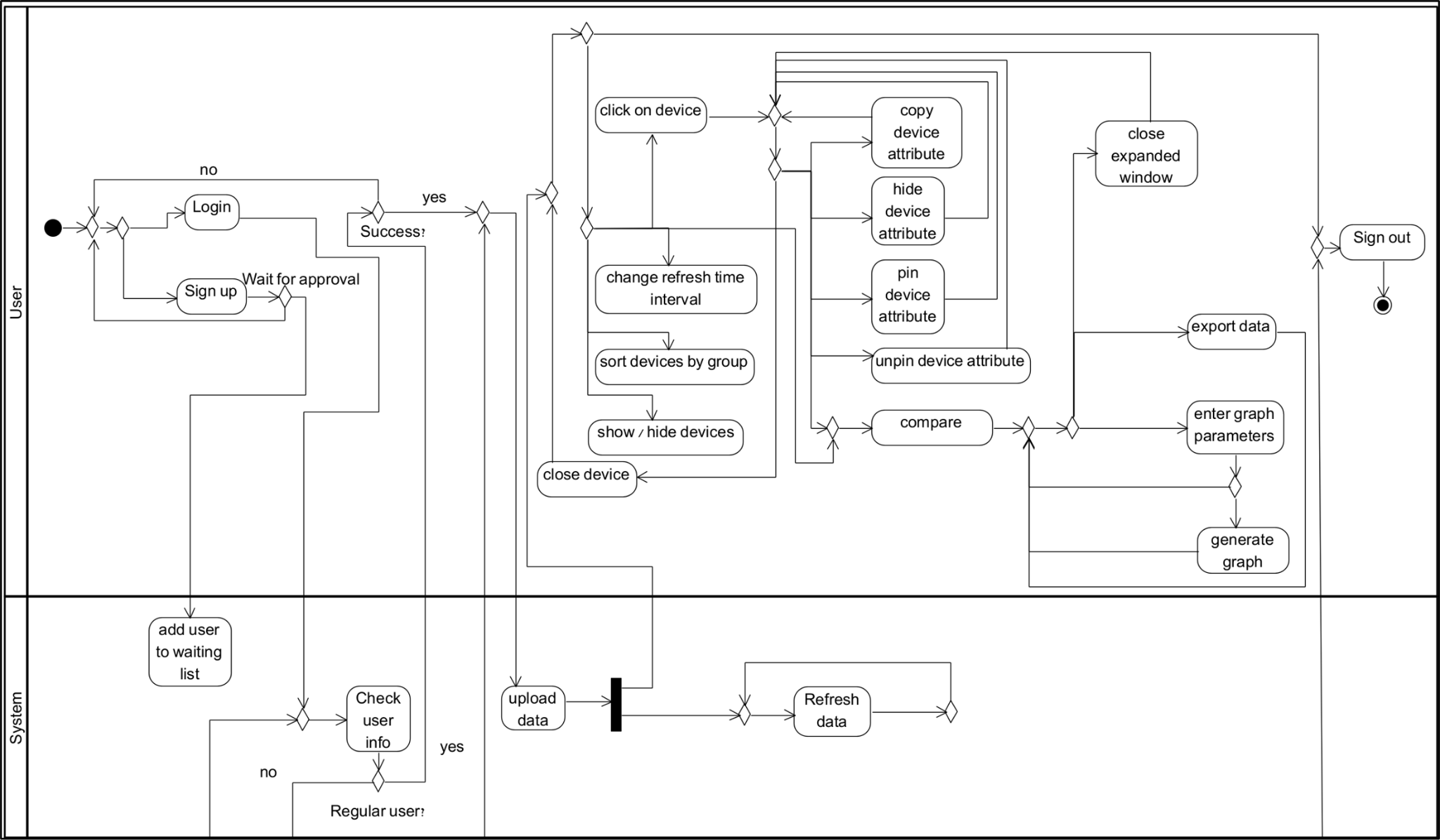
In the Use case diagram there are 2 players:

1. **Unapproved user** - player which can only sign up for the website and needs to wait until gets approved by the supervising professor.
2. **Approved user** - player which can sign in and do all the regular operations in the website including choosing the device from the map, clicking and watching all of its attributes. Moreover the player can copy the attribute, pin it in the list of attributes in the device details screen and generate a graph. The player can generate a graph with the last 100 samples or with specific arguments and can specify the graph type he wants to generate.

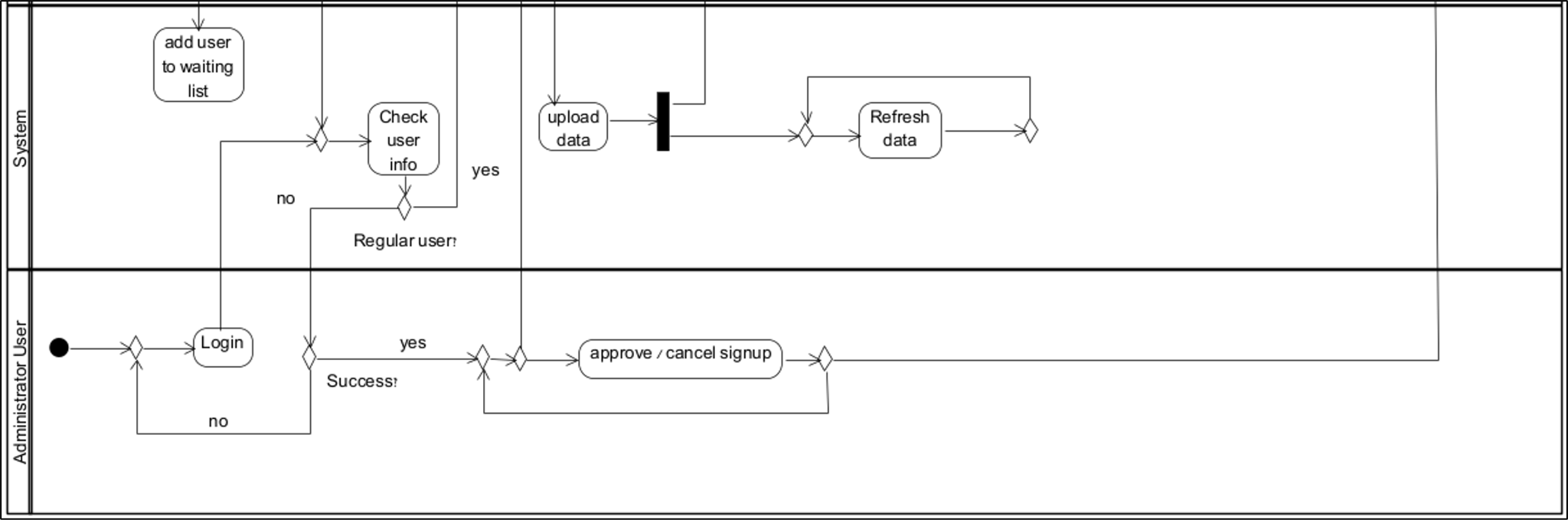


*figure 3. UseCase diagram*

#### 5.3 Activity Diagram

****

*figure 4a. Activity diagram - User and System*

****

*figure 4b. Activity diagram - Administrator user and System*

### 6. Evaluation

The evaluation of the project has been conducted using the System Usability Scale (SUS) questionnaire, accompanied by open-ended questions to gather detailed feedback for improvements. The evaluation involved two distinct groups of participants:

* A **pilot** group consisting of 33 students.
* An **experiment** group consisting of 29 students.

Both groups were presented with the same questionnaire and evaluation criteria, allowing us to gather comprehensive data about the system's usability from different user perspectives. This dual-group approach provided valuable insights into how different users interact with and perceive the system.

The SUS questionnaire was chosen for its reliability and validity in measuring system usability, while the open-ended questions allowed participants to provide detailed feedback about their experience and suggestions for improvement. This comprehensive evaluation approach not only helped us assess the overall usability of the system but also identified specific areas for enhancement. It has been presented to students who are enrolled in relevant courses, ensuring that feedback is collected from individuals with varying levels of experience in the subject. This thorough evaluation process guided the final adjustments to the system.

#### 6.1 Feedback Analysis

While the SUS scores provided a quantitative measure of usability improvements, the open-ended questions offered deeper insights into users' experiences, challenges, and suggestions for further enhancement. Participants were encouraged to share their thoughts on what they found intuitive, which aspects needed improvement, and any additional features they wished to see in the system. As a result, we categorized the participants’ thoughts into 2 main themes: **Positive feedback** and **Areas for improvements**.

#### 6.1.1 Positive feedback

Users appreciated the system’s responsiveness and efficiency, with several participants highlighting the website’s fast loading times and smooth interactions. Additionally, the UI design was described as informative and well-structured, contributing to an intuitive user experience.

This indicates that the system successfully aligns with usability principles, offering a user-friendly interface.

#### 6.1.2 Areas for Improvements

Several participants suggested enhancements to improve the system’s usability further:

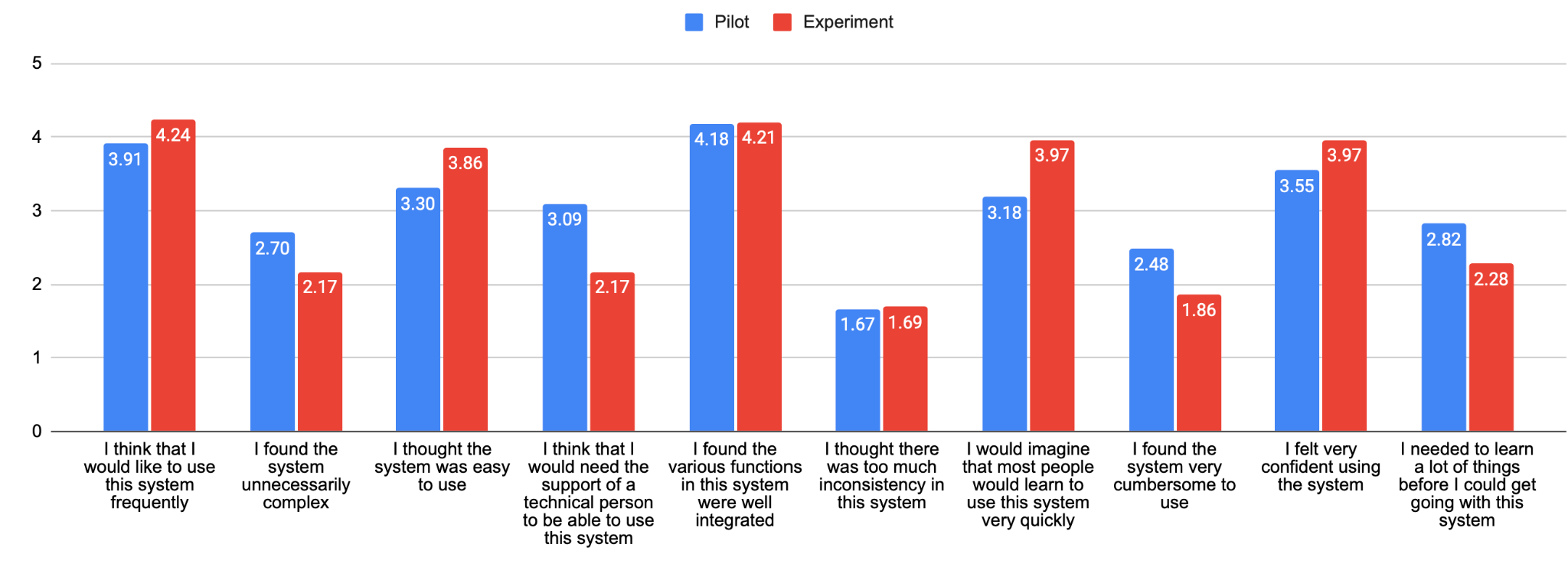
* **More explanations in different sections:** Users requested additional guidance or descriptions to clarify various features and functionalities.
* **Tutorial integration:** A step-by-step tutorial or onboarding guide was recommended to help new users navigate the system more effectively.
* **Visual alerts for device’s attribute selection:** Some users suggested implementing a clearer visual indicator when selecting a device to enhance feedback and interactivity.

These insights were instrumental in refining the system to better meet user expectations. By addressing these concerns, we aimed to create a more accessible and user-friendly platform that aligns with both usability standards and participant feedback.

#### 6.2 SUS Survey summary

The evaluation of the project has been conducted using the System Usability Scale (SUS) questionnaire, accompanied by open-ended questions to gather detailed feedback for improvements. The evaluation involved two distinct groups of participants: a pilot group consisting of 33 students who completed the evaluation first, followed by an experiment group of 29 students who participated 2 days later. This strategic timing allowed us to implement immediate improvements based on the pilot group's feedback before conducting the experiment group's evaluation.

The summary of the questionnaire from the 2 groups is presented in the following figure:



*figure 5. SUS questionnaire summary*

The SUS survey in the pilot group yielded a score of **63.4** while in the experiment group yielded a score of **75**. The rapid turnaround in implementing feedback from the pilot group demonstrated the project's agile approach to user-centered design, resulting in measurable improvements in the system's usability metrics within a short timeframe.

### 

### 7. Conclusion and Summary

This project aimed to develop a comprehensive IoT monitoring system for TAMK’s robotics laboratory, addressing the challenge of real-time data collection, visualization, and analysis from autonomous robotic units equipped with diverse sensors. This phase provided a structured framework for designing a system capable of handling live sensor data efficiently. With the knowledge gained from Phase A, Phase B focused on implementing these concepts into a fully functional web-based monitoring system.

We successfully developed a scalable and interactive platform that enables users to track and analyze sensor data in real time. The system's usability was validated through extensive acceptance testing and user evaluations, confirming that we effectively translated theoretical insights into a practical, working solution. This achievement not only enhances TAMK’s research and educational capabilities but also serves as a proof of concept for inter-institutional IIoT collaborations, demonstrating the potential for future advancements in robotics and smart manufacturing.

#### 7.1 Achievements

1. **Seamless API integration:** We established a reliable connection to external APIs, despite the need for VPN use, which ensures uninterrupted data access using DigitalOcean [26] platform to run the VPN alongside the server side.
2. **Demonstration of Robust IoT and Data Monitoring Capabilities:** We successfully demonstrated the system’s ability to monitor and process live data from robotics and sensors across various environments.
3. **Effective Use of Agile Methodology:** Successfully applying Agile principles, including regular feedback loops, sprints, and iterative improvements, to guide project development.
4. **User-Friendly Interface:** The product provides an intuitive interface that allows users to easily monitor, manage and visualize the data collected from connected devices.
5. **Scalable Architecture:** The system is designed to support increasing numbers of devices, ensuring scalability without compromising performance.
6. **High Data Accuracy:** The website guarantees that all sensor data is transmitted and displayed with high precision, providing users with reliable and accurate information.

#### 7.2 Challenges and Solutions

Throughout the development process, we encountered several challenges:

1. **Communication Barriers:** Team members are located in different countries, which makes communication a challenge. Misunderstandings or delays in communication can hinder progress.

To deal with this challenge:

* We used an asynchronous communication tool such as Teems with organized channels for different topics.

1. **Adaptation:** Changing from traditional CSS to Tailwind CSS to create a more adaptive website for smartphone access.

To deal with this challenge:

* We started with a component-by-component migration rather than changing everything at once.
* We used Tailwind’s responsive modifiers (sm:, md: lg:) to handle different screen sizes efficiently.

1. **Connection stability**: Relying on a VPN to access APIs from a different country leads to slower, unstable connections and complex configurations which impact the project's performance.

To deal with this challenge:

* We placed both the OpenVPN application and our server side in the same droplet in DigitalOcean to minimize latency between them.
* We set proper keep-alive settings in the OpenVPN configuration file to maintain stable connections.

### 8. Testing Plan

#### 8.1 Scope

This scope encompasses manual acceptance testing of user interfaces and workflows to ensure the system meets user requirements. The testing process involved evaluating functionality, usability, and performance through real-world interactions, identifying any issues, and refining the system based on user feedback.

#### 8.2 Objectives

1. Ensure the system meets all specified functional and non-functional requirements.
2. Identify and document any bugs, issues, or areas for improvement.
3. Validate the system’s reliability and stability over extended periods of use through real-world testing scenarios.

#### 8.3 Testing Approach

Our testing approach focused on acceptance testing to verify that the system meets user requirements and functions as expected in real-world scenarios. We conducted manual acceptance tests, where real users interacted with the system to validate usability, functionality, and performance.

The acceptance tests covered key workflows, ensuring that all features operated correctly and met user expectations. Feedback gathered from these tests helped us identify usability improvements, refine the user interface, and enhance the system’s overall stability. This hands-on approach ensured that the system was evaluated in realistic conditions, providing valuable insights that guided final adjustments before deployment.

#### 8.4 Test Cases

| **#** | **TestID** | **Precondition** | **Expected Result** | **Description** | **Test Result(Pass/Not Pass)** |
| --- | --- | --- | --- | --- | --- |
| **1** | SuccessLogin | The user approved by administrator, the user’s data exists in the database | The user is logged in and sees the main screen | 1)User inserts his userName.  2)User inserts his password.  3)User clicks the “Login” button. | Pass |
| **2** | FailLoginNotApproved | The user is not approved by the administrator | An error message is displayed - “Wait for approval” | 1)User inserts his userName.  2)User inserts his password.  3)User clicks the “Login” button. | Pass |
| **3** | FailLoginWrongPassword | The user inserts wrong password | An error message is displayed - “Wrong password” | 1)User inserts his userName.  2)User inserts his password.  3)User clicks the “Login” button. | Pass |
| **4** | SuccessSignIn | The user’s data doesn’t exist in the database | An message is displayed - “Complete, wait for approval” | 1)User inserts userName.  2)User inserts password.  3)User clicks the “Sign-in” button. | Pass |
| **5** | FailSignInUserNameTaken | The user’s data exists in the database | An error message is displayed - “UserName already exist” | 1)User inserts userName.  2)User inserts password.  3)User clicks the “Sign-in” button. | Pass |
| **6** | FailSignInMissingData | The user didn’t fill in all necessary information. | An error message is displayed - “Please insert username and password” | 1)User inserts password.  2)User clicks the “Sign-in” button. | Pass |
| **7** | ChoosingDevice | 1)The user login successfully.  2)The data was fetched successfully. | Device attribute screen will update with the data | User clicks on some device in the map | Not Pass - Pilot group  Pass - Experiment group |
| **8** | SortDisplay | 1)The user login successfully.  2)The data was fetched successfully. | The map will update and show the chosen devices. | User clicks on the devices in Device screen he wants to show on map | Pass |
| **9** | SuccessApproveUser | 1)The user login successfully.  2)The type of user is “Administrator”.  3)The data of pending users is fetched successfully. | An message is displayed = “Approved user successfully” | Administrator clicks on “approve” button of some pending user in the Pending screen | Pass |
| **10** | SuccessDenyUser | 1)The user login successfully.  2)The type of user is “Administrator”.  3)The data of pending users is fetched successfully. | An message is displayed = “deny user successfully” | Administrator clicks on “deny” button of some pending user in the Pending screen | Pass |
| **11** | SuccessCreateGraph | 1)The user login successfully.  2)Data was fetched successfully | An updated graph will appear in the extended window | 1)User clicks in the device attribute screen on attribute.  2)User clicks on the compare window button.  3)User inserts a range of time and date.  4)User clicks on the “Submit” button. | Not Pass - Pilot group  Pass - Experiment group |
| **12** | SuccessOperational | 1)The user login successfully.  2)Data wasn’t fetched successfully | The website will be empty but operational. | The website is running and operational although there’s no incoming data from the server. | Pass |

*table 3. Test cases*

### 9. AI Tools and Prompts

During our work on part a, we used AI tools such as: ChatGPT [23], Claude [24] and Copilot [25]. We used these different AI tools in a variety of areas such as checking syntax and spelling, conducting literature introduction research as well as in the practical part to check on various technologies and methods.

#### 9.1 Prompts Used

**Prompt 1: “**We develop an integrated internet of things software system that monitors robotics data, using various sensors that are on the robotics and visualizes that data. Please suggest an appropriate name for such a site”.

**Answer:** ChatGPT suggested different kinds of names for such systems including: “RoboSense”, “MonitorBot”, “IoT Sphere”, “RoboMo” and such more. Eventually we decided to choose the name “RoboMo” which stands for: Robotics Monitoring.

**Prompt 2: “**We develop an integrated internet of things software system that monitors robotics data, using various sensors that are on the robotics and visualizes that data. Please provide relevant literature about IoT,IIoT, robotics and monitoring data”.

**Answer:** ChatGPT and Claude provided us with different types of articles that connected to the subjects we mentioned. This helped us in writing the literature review and finding more suitable terms that are also connected to our project.

**Prompt 3:** “We want to use better server-client communication in our React project, please list the best ways to achieve this”.

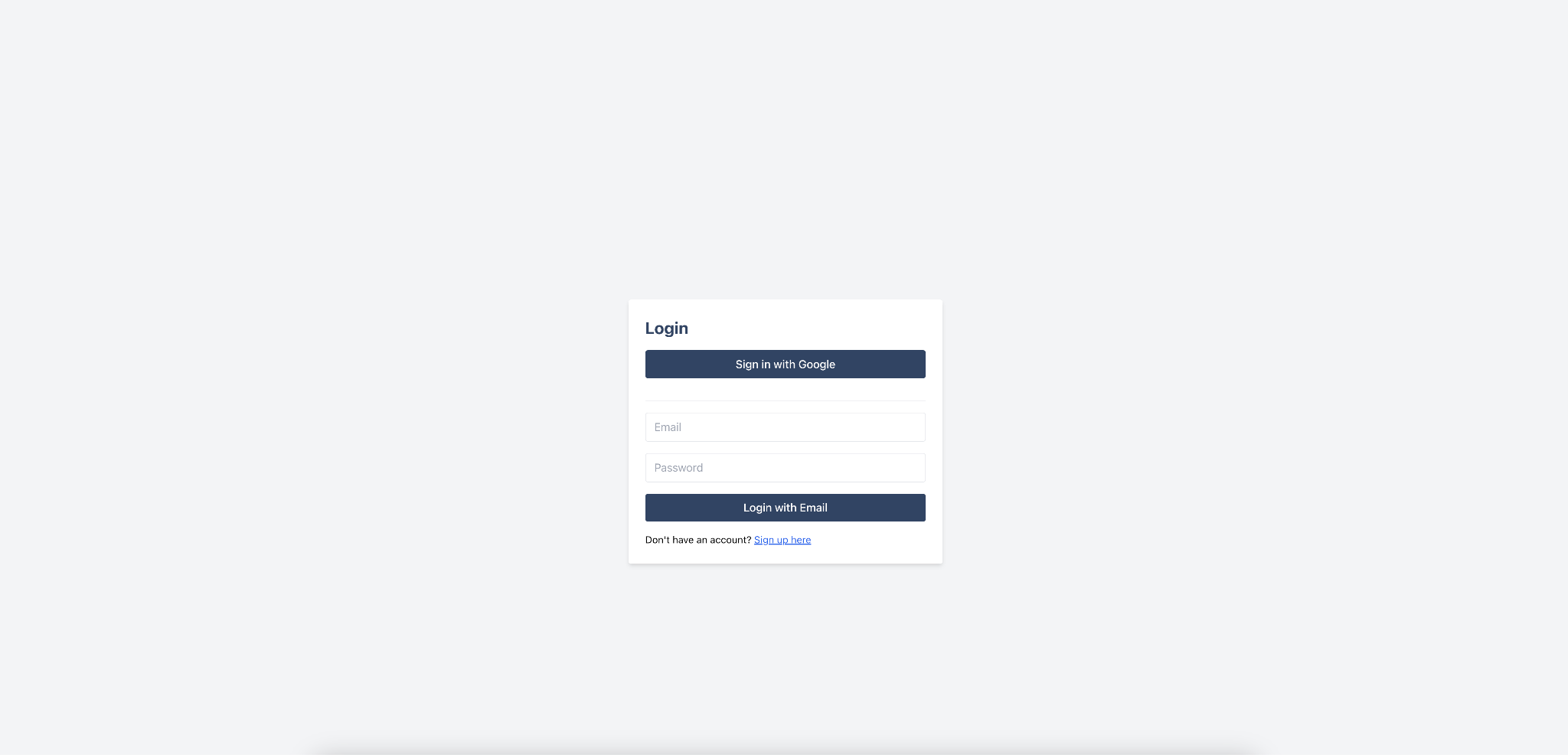
**Answer:** ChatGPT provided us several approaches that we can choose depending on our use case, project requirements and scalability needs. Among these approaches chatGPT introduced WebSocket, which we chose eventually, and explained the way it works. ChatGPT explained that it enables full-duplex communication between the client and the server and that it is ideal for real-time, bidirectional communication and live updates and explained in detail about the advantages of this approach.

### 10. User Guide

#### 10.1 Login Page

1. **Purpose:** This is the main landing page of the RoboMo website. It offers the user the option to login to the system with email and password, or login to the system using Google. There are 3 options for doing this on this page: login with email and password, login with Google and Sign up for new users.
2. **Key Elements:**

* **“Sign in with Google”** Button: The button allows the user to sign in using Google authentication.
* **“Login with email”** button: The button allows the user to login using email and password which he signed up with before.
* **“Sign up here”** button: The button links the user to the registration page (figure 7).

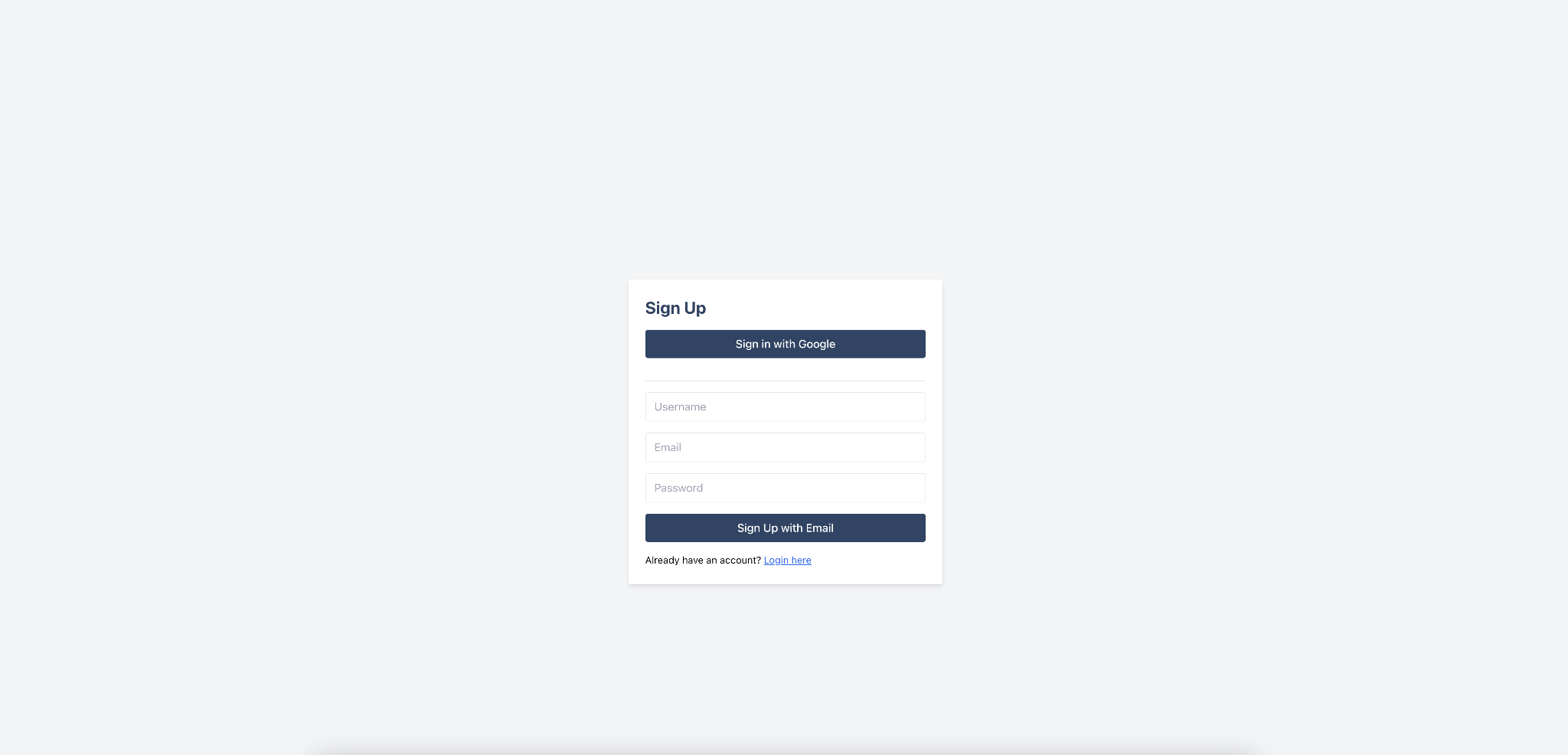


*figure 6. Login screen*

#### 10.2 Sign Up Page

1. **Purpose:** This is the page where the user first signs up for the website.
2. **Key Elements:**

* **“Sign in with Google”** button: The button allows the user to sign up using Google authentication.
* **“Sign Up with Email”** button: The button allows the user to sign up using email and password. After clicking the sign-up button, the user will be asked to confirm the sign-up by clicking a link that is sent to the email used during registration.
* **“Login here”** button: The button directs the user to the login page (figure 6).

**

*figure 7. Sign up screen*

#### 10.3 Homepage

1. **Purpose:** This is the main page of the website, it displays the connected devices. All the interactions of the user occur on this screen. The homepage consists of 6 components: Widget bar, Devices, Configuration, Map, Device attribute and Compare screen.
2. **Key Elements:**

* **“Signout”** button: The button disconnects the user and directs him to the login screen (figure 6) .



*figure 8. Homepage*

#### 10.4 Widget bar

1. **Purpose:** The widget bar informs the user about important things on the website. It allows the user to log out from the website, view a specific attribute data, view the amount of connected users, amount of active devices and the amount of data in bytes that are transferred from the server.
2. **Key Elements:**

* **UserName (Or Benyamin)**: Displays the currently logged-in user’s name. It indicates the time the user connected to the website.
* **Pin attribute (N CO 5 Value)**: Displays the data of the sensor the user chose to pin to the widget bar. It shows the name of the sensor (N CO 5) the device id it linked to (SPS:30:001) and the current value of the sensor (6.23). When clicking on the pin button it unpins the current attribute from displaying in the widget bar.
* **#d Connected Users**: Displays the amount of users that are connected to the website right now.
* **#d Active Devices**: Displays the amount of devices that are available and shown on the map (figure 12.a).
* **#f Bytes/s:** Displays in text and graph the amount of bytes that transferred from the server every second.

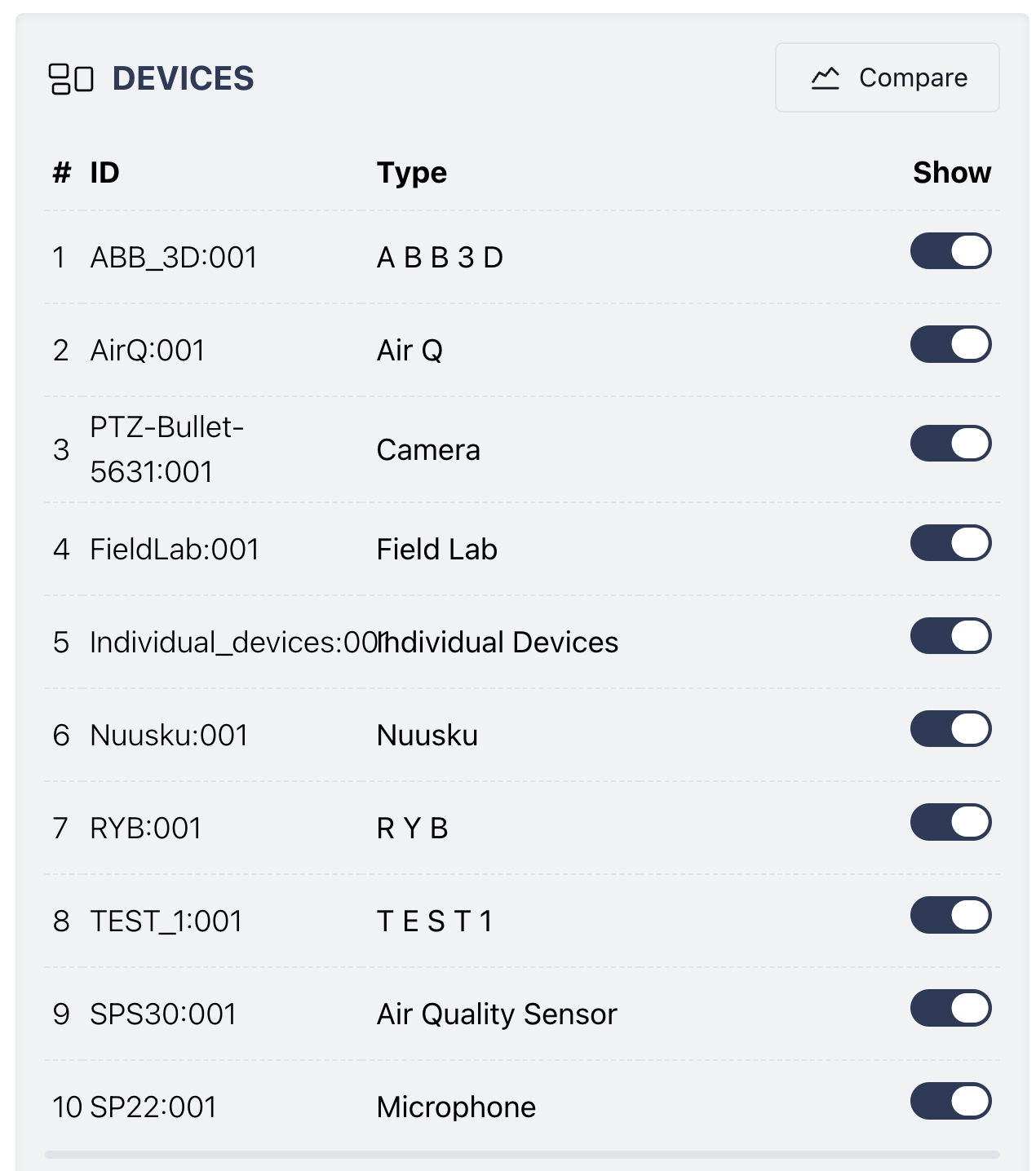


*figure 9. Widget bar*

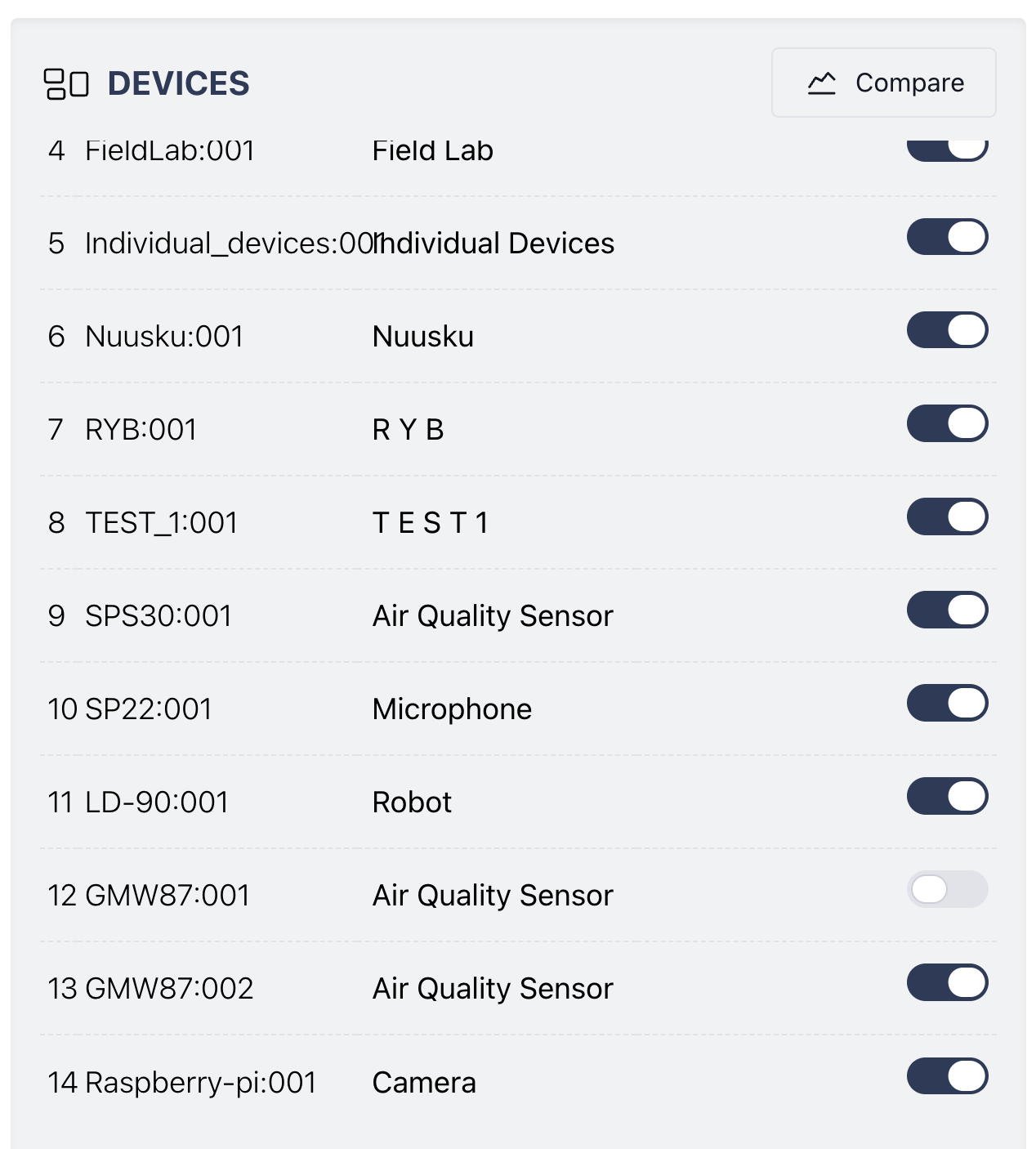
#### 10.5 Devices list

1. **Purpose:** The devices list shows the current devices that have been displayed on the map. It shows for each device its id, type and option to show or hide it from displaying on the map.
2. **Key Elements:**

* **“Show”** button: The button allows users to choose which device to show on the map. When the “Show” button is in color, the device is shown on the map, and when it is grey, the device is not shown on the map (figure 12.b).
* **“Compare”** button: The button opens the compare window and allows users to create graphs using the sensor's data (figure 14.a).
* Each device line allows users to click and view the chosen device’s attributes in the device attribute section (figure 13).



*figure 10.a. Devices list*

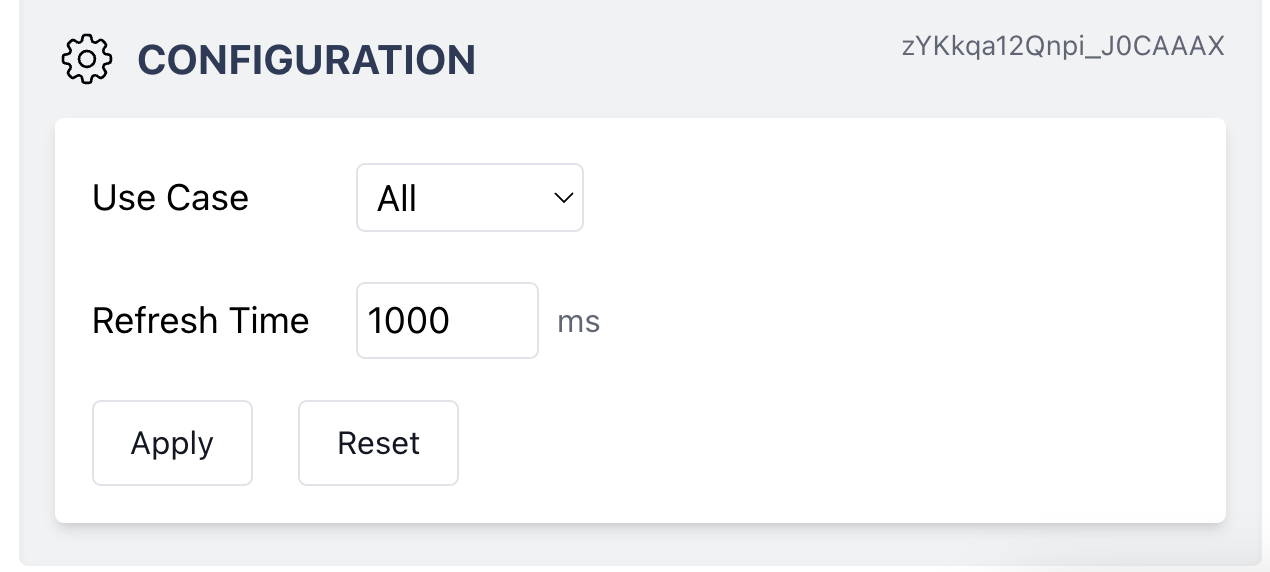
**

*figure 10.b. Devices list with one device not shown on the map (Device number 12)*

#### 10.6 Configuration section

1. **Purpose:** The configuration section allows users to sort the devices displayed on the map and adjust the refresh time speed.
2. **Key Elements:**

* **“Use Case”** Drop Down List: It allows users to sort the devices displayed on the map by use case value.
* **“Refresh Time”** text box: It allows users to adjust the website’s data refresh rate.

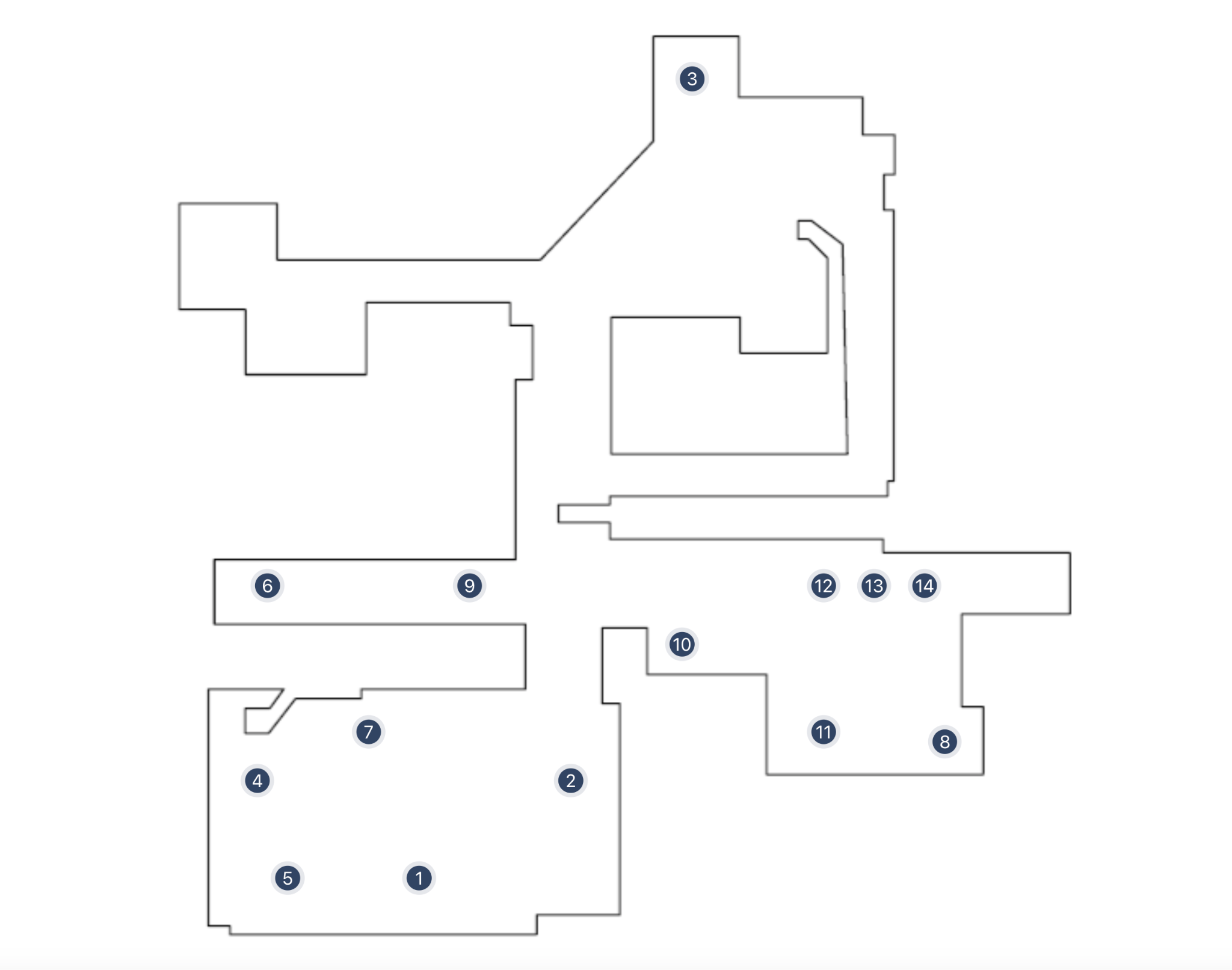
**

*figure 11. Configuration section*

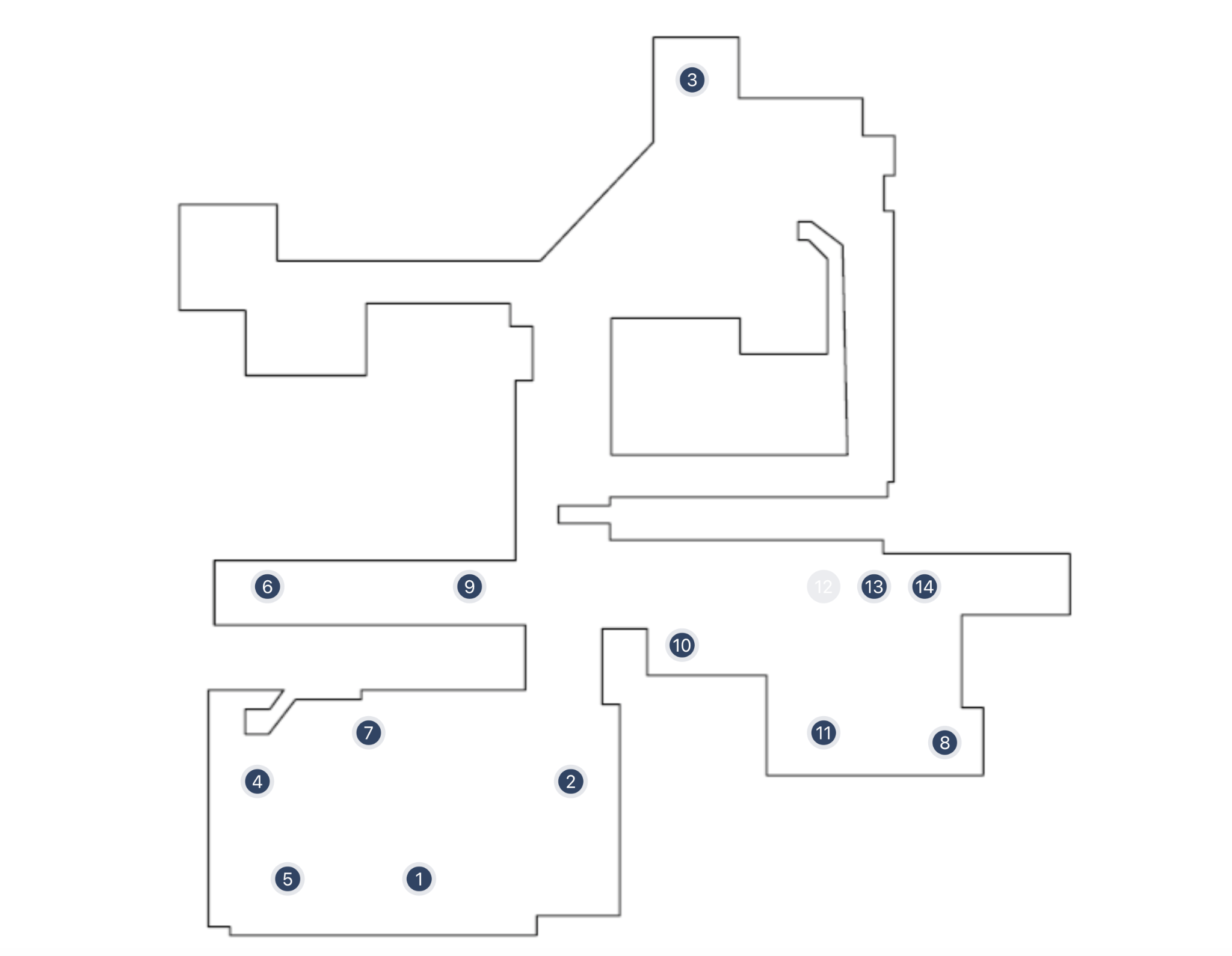
#### 10.7 Map

1. **Purpose:** The map allows the users to view the location of the robots throughout the laboratory.
2. **Key Elements:**

* **“#d”** icon: The icons allow users to click and view the chosen device’s attributes in the device attribute section (figure 13.a).
* Once the device is set to not be shown on the map, the icon will appear in grey (figure 12.b).

**

*figure 12.a. Map with all devices shown on the map*

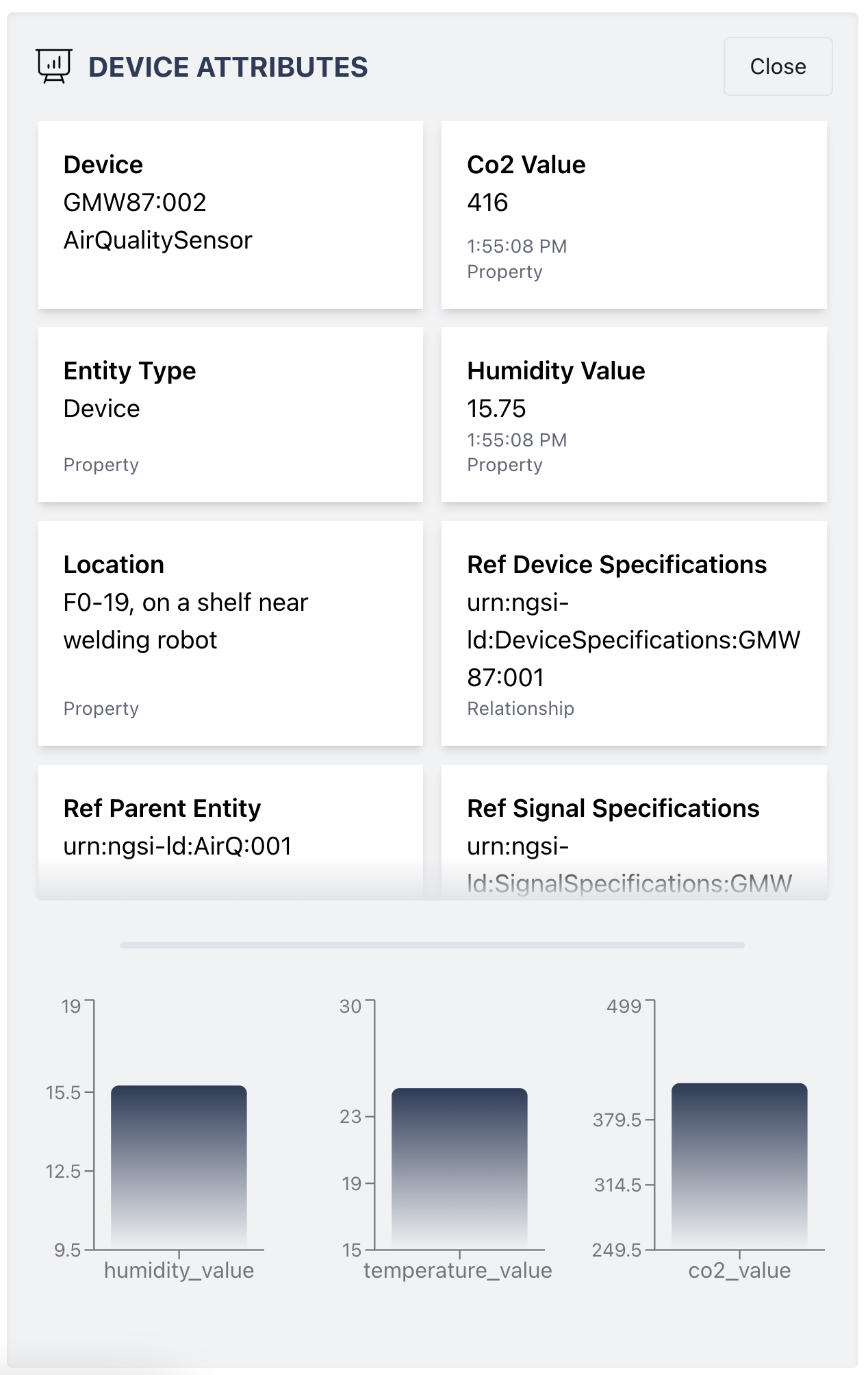
**

*figure 12.b. Map with one device not shown on the map (Device number 12)*

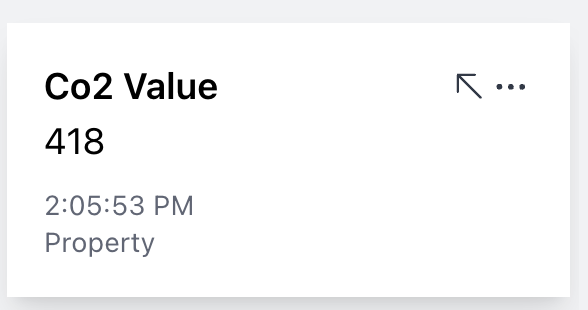
#### 10.8 Device attribute

1. **Purpose:** The device attribute section allows users to view the chosen device’s data. In the bottom of the section there are a few graphs that display the sensor's value.
2. **Key Elements:**

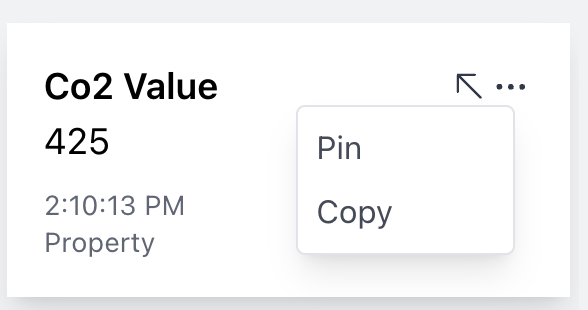
* **Attribute**: On each attribute there appears the name of the sensor, the value and the time of the last update.
* **“Close”** button: The button closes the view on the chosen device and its attributes. The users will return to the website as in the beginning (figure 8).
* **Attribute options**: When hover with mouse on attribute there are 2 options available (figure 13.b):
  + **Arrow**: Opens the compare window with the information of the chosen attribute (figure 14.b).
  + **3 Dots**: when clicking on the 3 dots as in figure 12.c the users will be able to copy the attribute value or pin it to the widget bar (figure 9).



*figure 13.a. Device attribute*

**

*figure 13.b. Device attribute options*

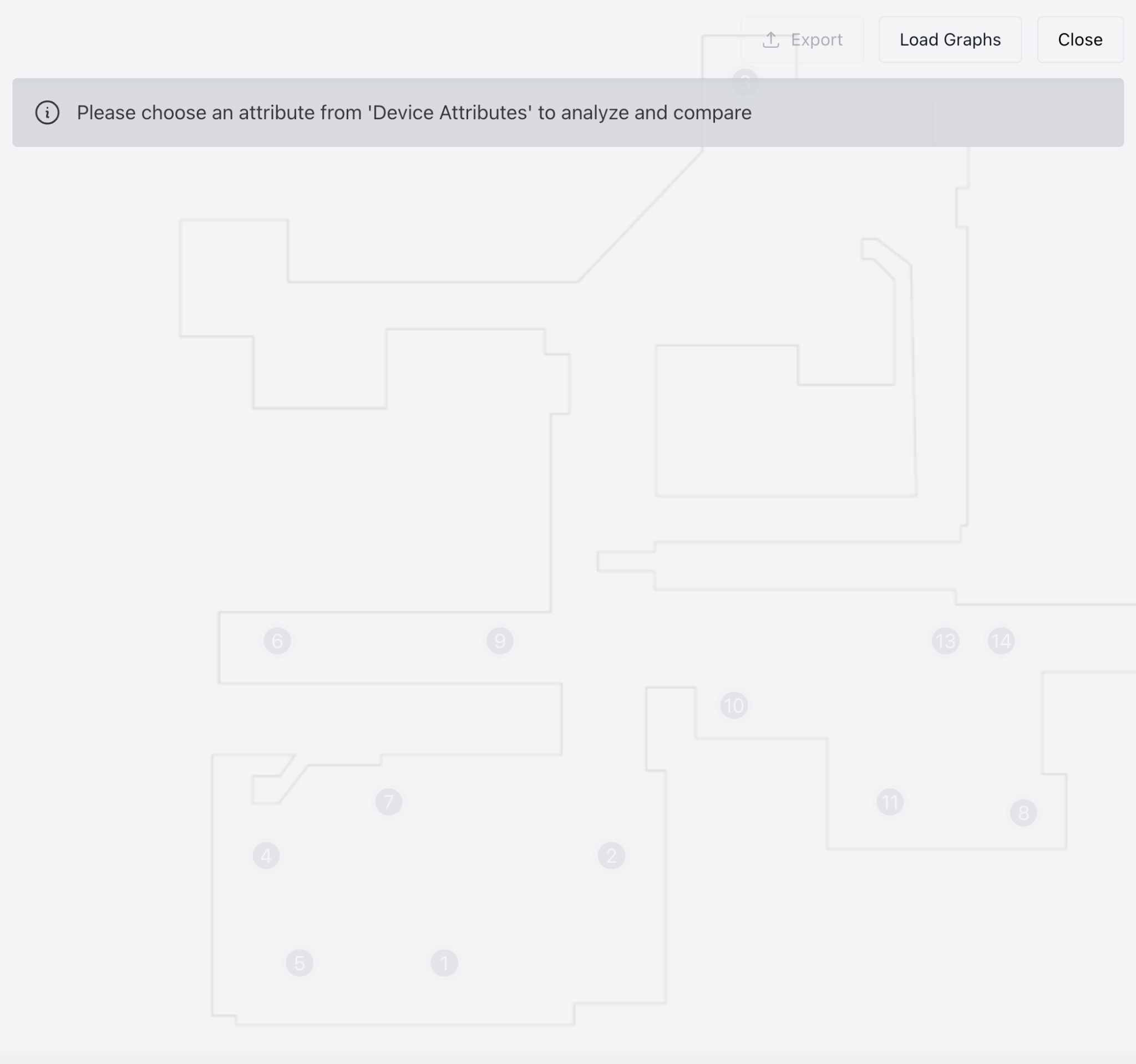
**

*figure 13.c. Device attribute options*

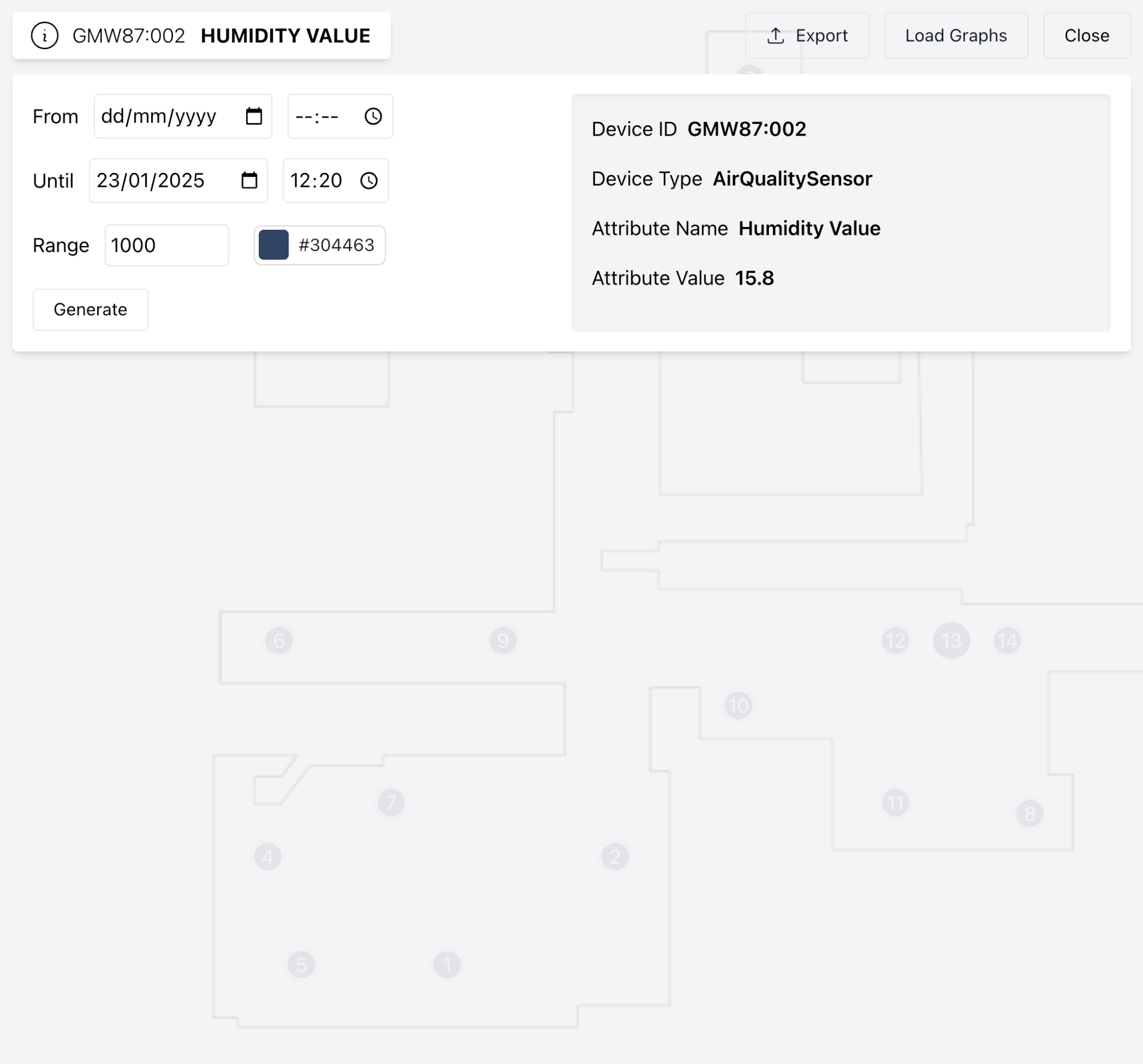
#### 10.9 Compare section

1. **Purpose:** The compare section allows users to compare between different devices’ sensors and view the sensors' values differences.
2. **Key Elements:**

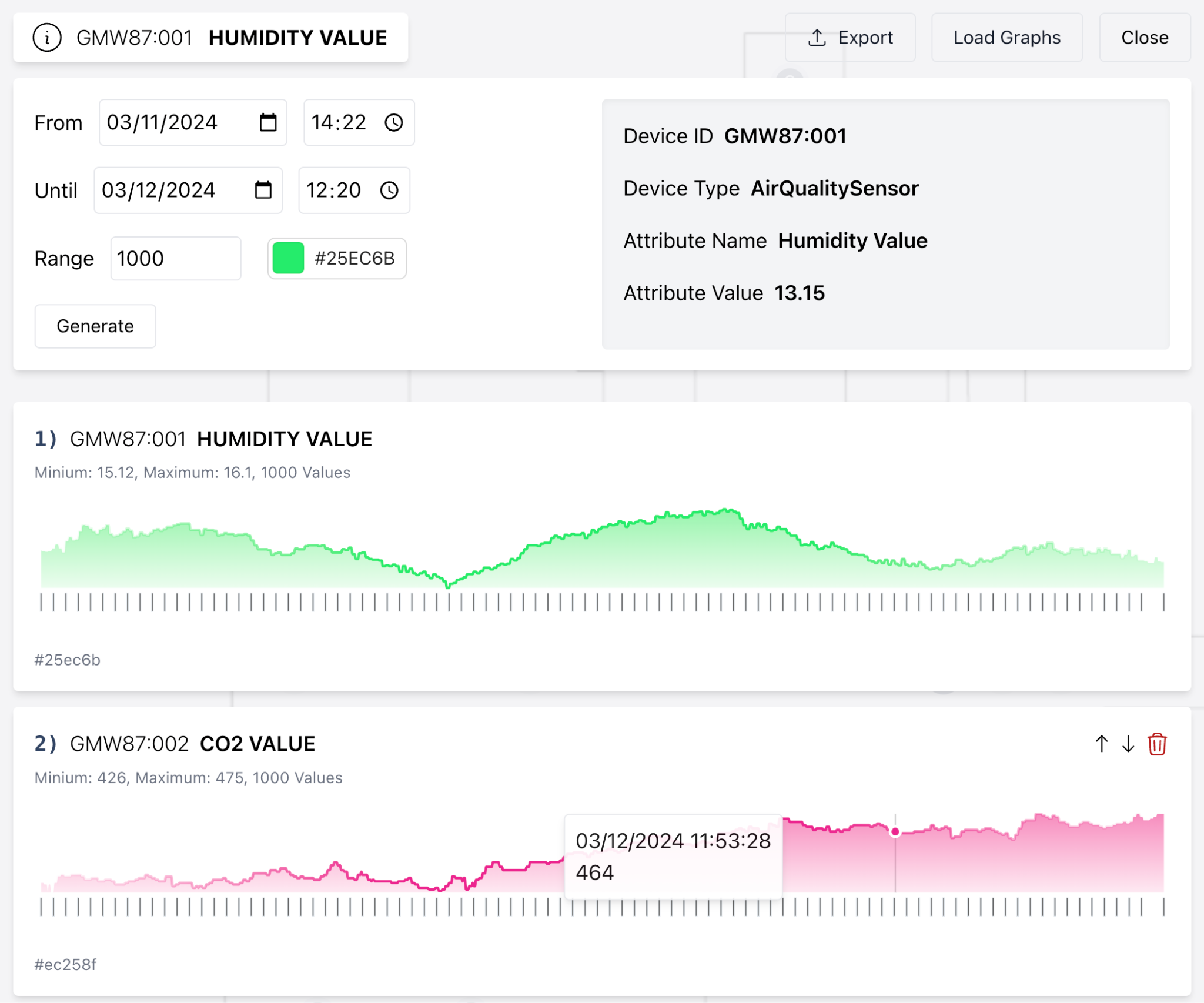
* **“Close”** button: The button closes the Compare section.
* **“Load Graphs”** button: The button allows users to load saved graphs to the window.
* **“Export”** button: The button allows users to export the graphs’ data to an Excel file.
* Once users choose device's attribute they will see the compare window as in figure 14.b.
* **“Generate”** button: The button generates the graphs by using the attribute id and the values of start date, end date, start time, end time, range (amount of data) and color of graph.
* Once the users load graphs or create graphs, they will see the compare window as in figure 14.c.
* When finished generating the graph and the compare screen is open, users can click on different devices from the device list, choose attributes and create another graph that will be displayed on top of the previous graph.
* When users hover with a mouse on the graph’s line, they will have the option to view the value and the time on the specific point (figure 14.c)
* When users hover with a mouse on an existing graph, they will have the options to move the graph in the view list up and down (depending whether there are more than 1 graph in the view) and delete the graph from the screen (figure 14.c).
* Each graph is saved automatically when created and can therefore be loaded back when needed unless users delete it.

**

*figure 14.a. Compare screen without choosing device*

**

*figure 14.b. Compare screen with choosing device attribute*

**

*figure 14.c. Compare screen with graphs*

### 11. Maintenance Guide

#### 11.1 Installation

#### 11.1.1 Prerequisites

* Node.js and npm (for server-side javascript, for client-side javascript, react with tailwind)
* Git
* Firebase (for database)
* Render (for client-side deployment)
* DigitalOcean (for server-side deployment)
* OpenVPN (for authentication with the server side - when running locally)
* Google account (for accessing the website when login or signup)

#### 11.1.2 Setup

1. **Clone Github repository:**



*figure 15. RoboMo repository*

1. **Install required modules:**
   1. Enter the client and server directories and run npm install on each one.
2. **Private credentials:**
   1. .env files: Add environment files with private credentials.
   2. OpenVPN config file: Add the openVPN config file.
   3. SSL certificates: Generate SSL certificates for HTTPS.

#### 11.2 Deploying the website

#### 11.2.1 Client side

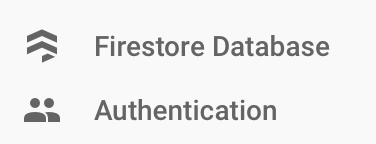
1. Have an account on Redner.
2. Choose “Static site”.
3. Set the repository as the repository for the static web.
4. Set the branch to main.
5. Set the “Root Directory” to “client” (as the name of the client directory).
6. Set “Build Command” to “npm run build”.
7. Set “Publish Directory” as “build”.
8. Set “Auto-Deploy” as “Yes”.
9. Deploy the “Static Site”.

#### 11.2.2 Server side

1. Have an account on “DigitalOcean”.
2. Create “Droplets” using image “Ubuntu 22.04 (LTS)”.
3. Choose a plan of 1GB RAM, 25GB SSD.
4. Create Droplet.
5. Connect to the created Droplet using SSH.
6. Install OpenVPN.
7. Transfer the OpenVPN config file.
8. Transfer the server side code.
9. Edit config file to run OpenVPN and server side simultaneously.
10. Buy Domain.
11. Enter the Domain in the Droplet.
12. Run the config file.

#### 11.3 Database

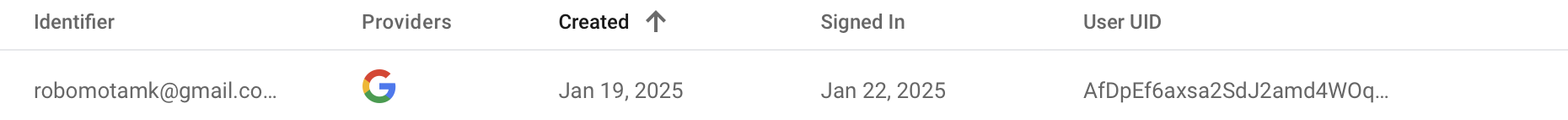
Firebase cluster with the following collections



*figure 16. Firebase collection*

#### 11.3.1 Authentication

For each user, the email, registration method, username, registration date, last login date and the user ID are saved.

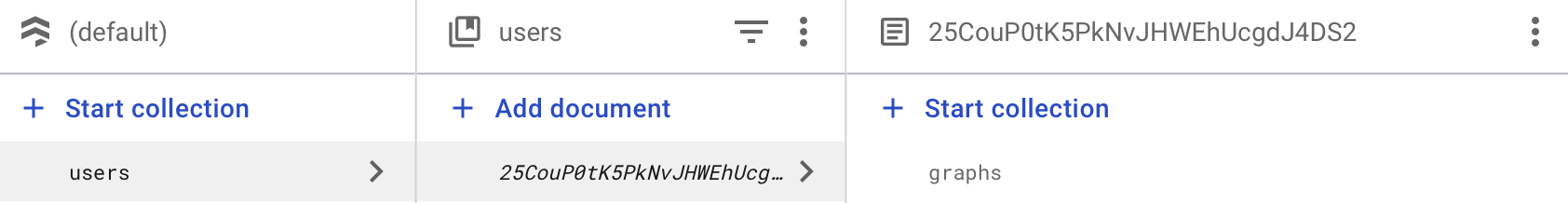


*figure 17. Authentication*

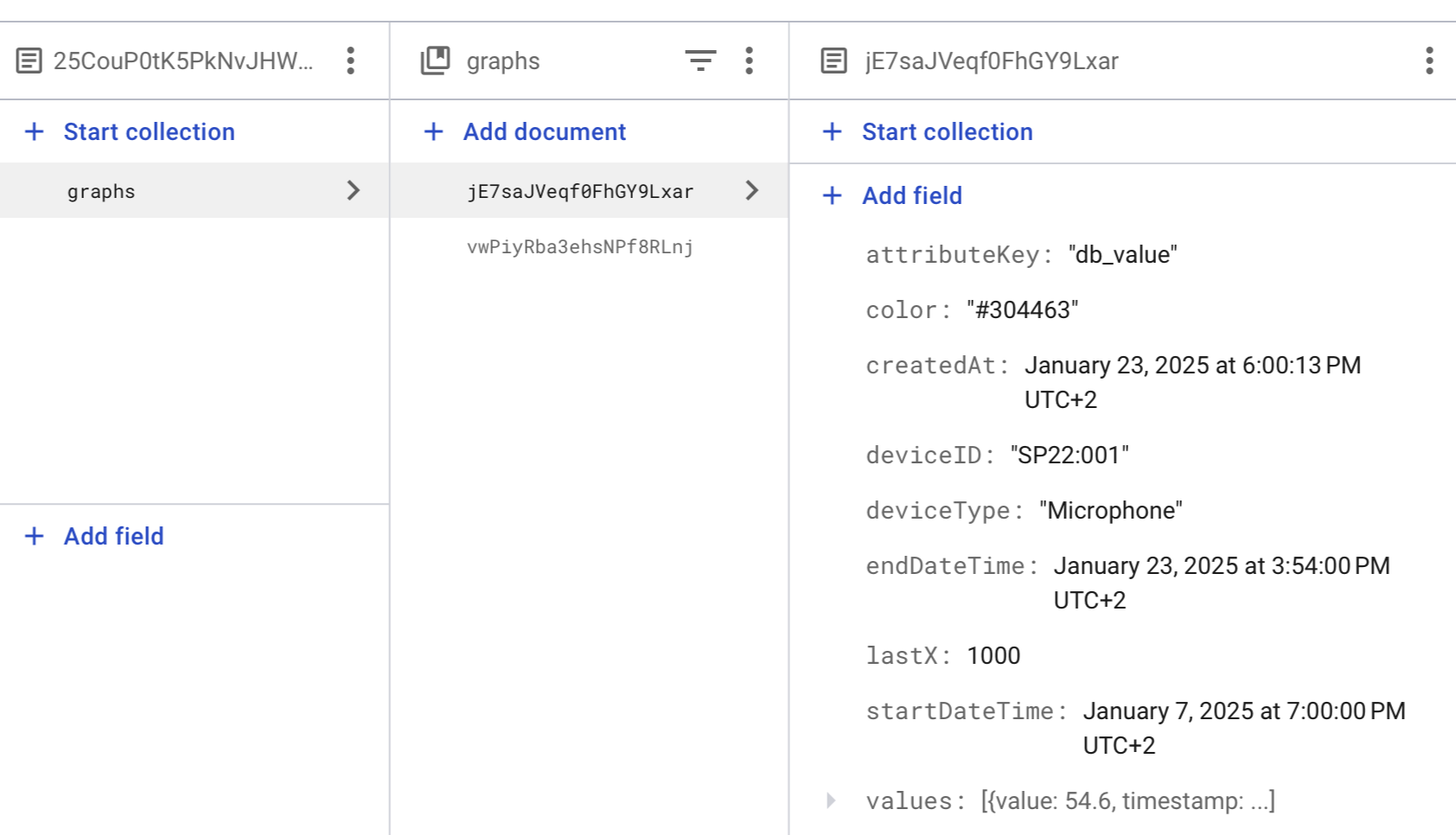
#### 11.3.2 Firestore Database

Graph data created in the Compare window are automatically saved to Firebase, with data stored only for the specific user based on their user ID (figure 18.a).

Each graph receives a unique graph id. Saved graph details include, the attribute key, graph’s color, creation date, device’s id, device’s type, start date, end date, start time, end time, amount of data used to create the graph and the values are saved (figure 18.b).



*figure 18.a. Firestore Database*

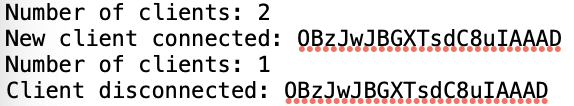
**

*figure 18.b. Firestore Database graph data*

#### 11.4 Server requests

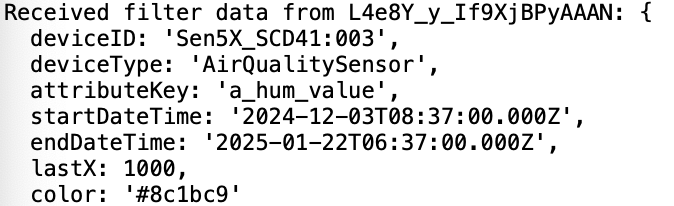
While users interact with the website, the server saves all logs and user requests.

For example, when a client connects to the website, the server updates the log with the number of users currently connected and logs the user who disconnected (figure 19).



*figure 19. 2 connected users and one user is logout*

When users generate graphs, the server updates the log with the request, including the graph's parameters and user ID (figure 20).



*figure 20. request for generating graph*

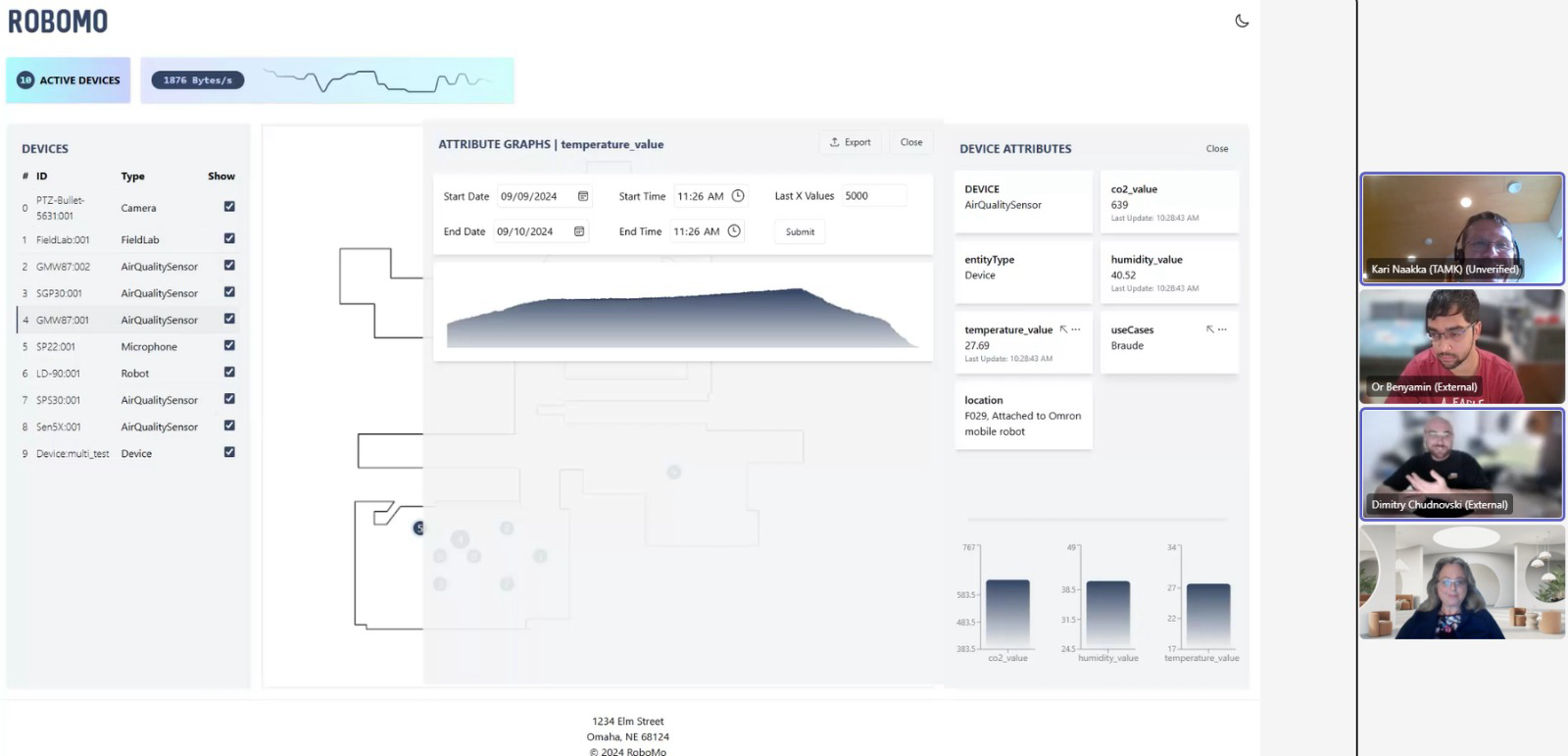
### 12. References

1. Demertzi, V., Demertzis, S., & Demertzis, K. (2023). An Overview of Privacy Dimensions on the Industrial Internet of Things (IIoT). Algorithms, 16(8), 378.
2. Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. IEEE access, 8, 108952-108971.
3. Grieco, L. A., Rizzo, A., Colucci, S., Sicari, S., Piro, G., Di Paola, D., & Boggia, G. (2014). IoT-aided robotics applications: Technological implications, target domains and open issues. Computer Communications, 54, 32-47.
4. Kamilaris, A., & Botteghi, N. (2020). The penetration of Internet of Things in robotics: Towards a web of robotic things. Journal of ambient intelligence and smart environments, 12(6), 491-512.
5. Khan, A., Aziz, S., Bashir, M., & Khan, M. U. (2020, March). IoT and wireless sensor network based autonomous farming robot. In 2020 international conference on emerging trends in smart technologies (ICETST) (pp. 1-5). IEEE.
6. Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A., & Qureshi, B. (2020). An overview of IoT sensor data processing, fusion, and analysis techniques. Sensors, 20(21), 6076.
7. Li, P., & Liu, X. (2019, July). Common sensors in industrial robots: A review. In Journal of Physics: Conference Series(Vol. 1267, No. 1, p. 012036). IOP Publishing.
8. Nayyar, A., Puri, V., Nguyen, N. G., & Le, D. N. (2018). Smart surveillance robot for real-time monitoring and control system in environment and industrial applications. In Information Systems Design and Intelligent Applications: Proceedings of Fourth International Conference INDIA 2017 (pp. 229-243). Springer Singapore.
9. Ray, P. P. (2016). Internet of robotic things: concept, technologies, and challenges. IEEE access, 4, 9489-9500.
10. Singh, M., Fuenmayor, E., Hinchy, E. P., Qiao, Y., Murray, N., & Devine, D. (2021). Digital twin: Origin to future. Applied System Innovation, 4(2), 36.
11. Vermesan, O., Bröring, A., Tragos, E., Serrano, M., Bacciu, D., Chessa, S., ... & Bahr, R. (2022). Internet of robotic things–converging sensing/actuating, hyperconnectivity, artificial intelligence and IoT platforms. In Cognitive hyperconnected digital transformation (pp. 97-155). River Publishers.
12. [React](https://react.dev/)
13. [What React Is and Why It Matters [Book]](https://www.oreilly.com/library/view/what-react-is/9781491996744/ch01.html)
14. [What Is JavaScript: A Beginner's Guide to the Basics of JS](https://www.hostinger.com/tutorials/what-is-javascript)
15. [Tailwind CSS: What It Is, Why Use It & Examples](https://blog.hubspot.com/website/what-is-tailwind-css)
16. [What Is Node.js? A Complete Guide for Developers](https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs)
17. [FIWARE, the standard that the IoT needs – TM Forum](https://www.tmforum.org/press-and-news/fiware-standard-iot-needs/)
18. [What is FIWARE?](https://www.fiware.org/2011/05/17/what-is-fiware/)
19. [Firebase Realtime Database](https://firebase.google.com/docs/database)
20. [IEEE SA - IEEE 828-2012](https://standards.ieee.org/ieee/828/5367/)
21. [Web Content Accessibility Guidelines (WCAG) 2.2](https://www.w3.org/TR/WCAG22/)
22. [What is web socket and how it is different from the HTTP? - GeeksforGeeks](https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/)
23. [ChatGPT](https://chatgpt.com/auth/login)
24. [Claude](https://claude.ai/login?returnTo=%2F%3F)
25. [Copilot](https://copilot.microsoft.com/chats/YqGoszd3F4fGVfQjZpqA9)
26. [DigitalOcean](https://www.digitalocean.com/)

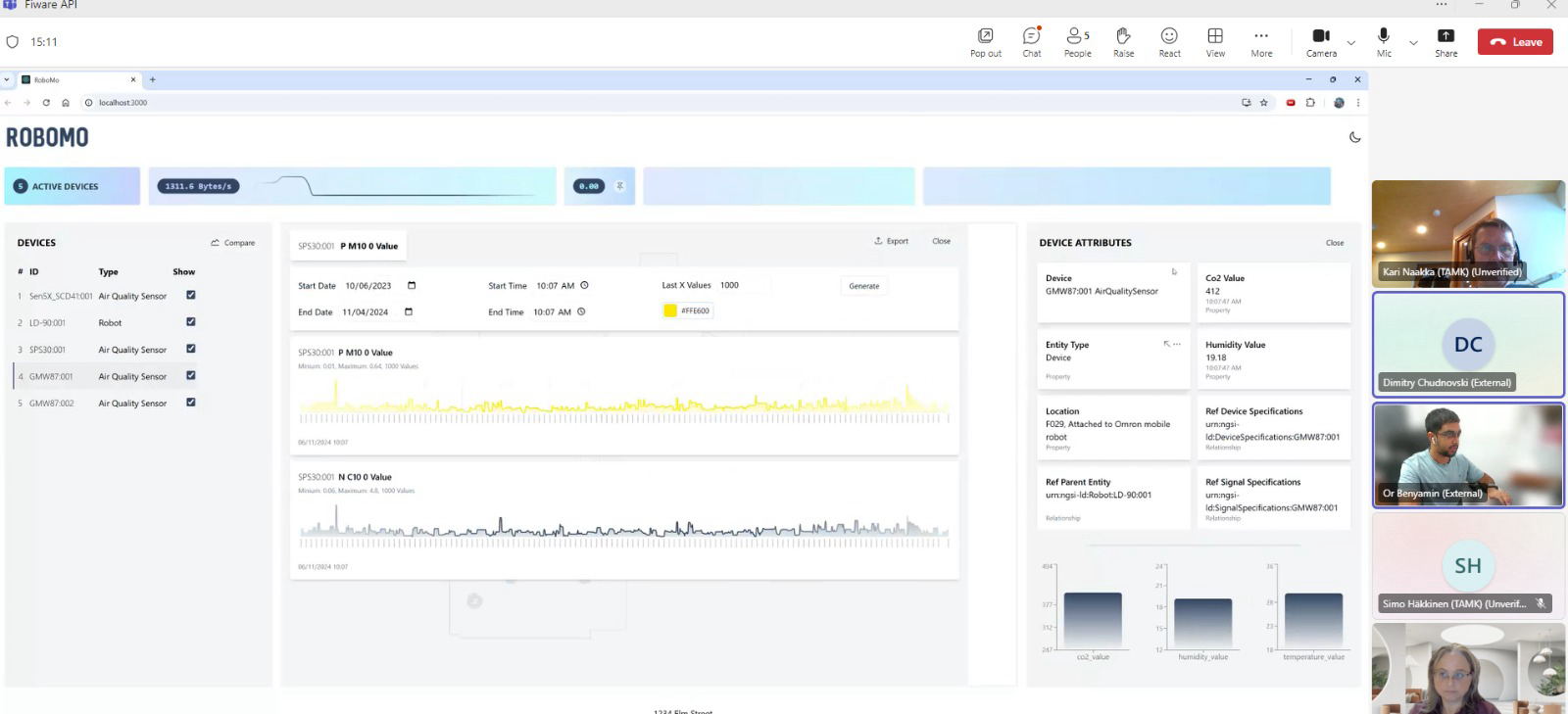
### 

### 13. Appendices

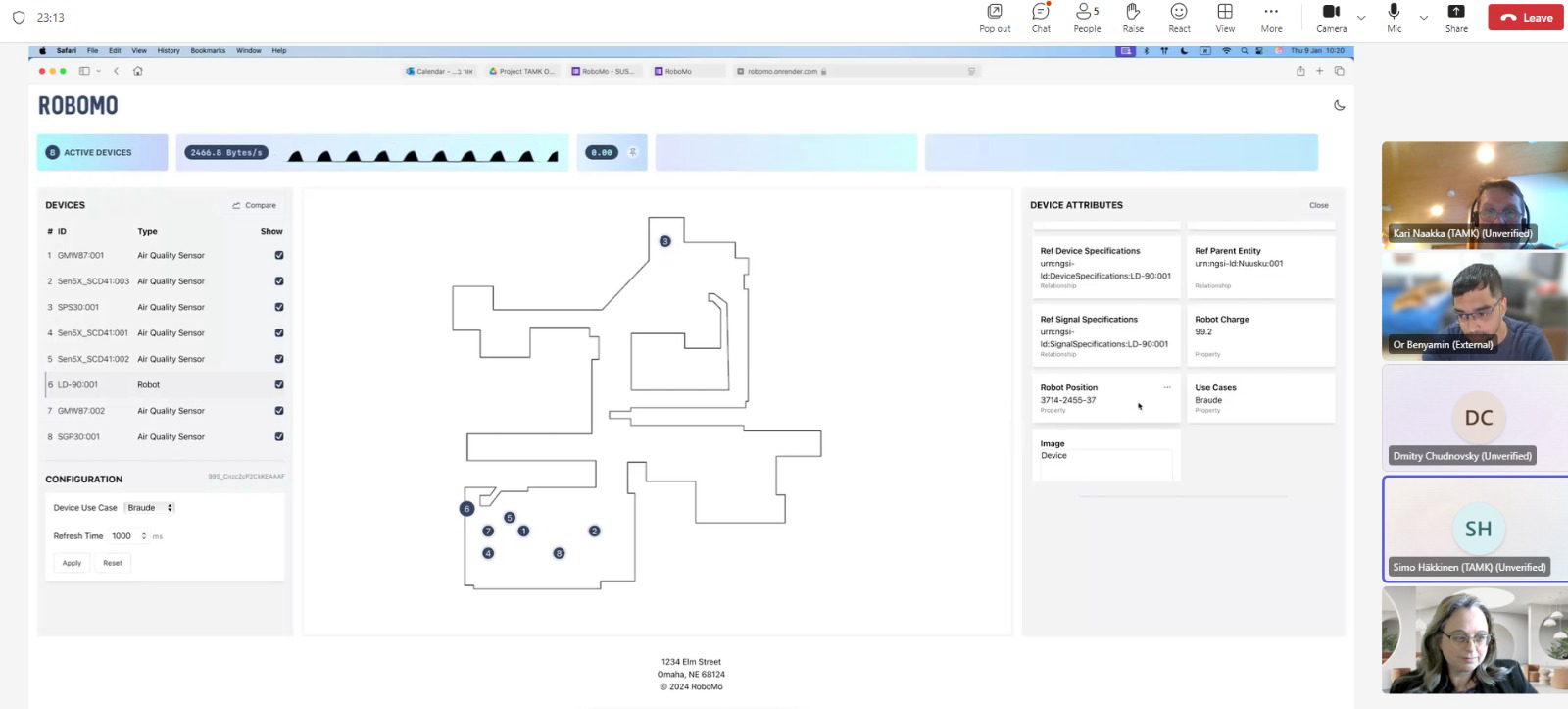
#### 13.1 Meetings Capture

****

*picture 1. 4th meeting with the customer in Teams*

****

*picture 2. 5th meeting with the customer in Teams*

****

*picture 3. 7th meeting with the customer in Teams*