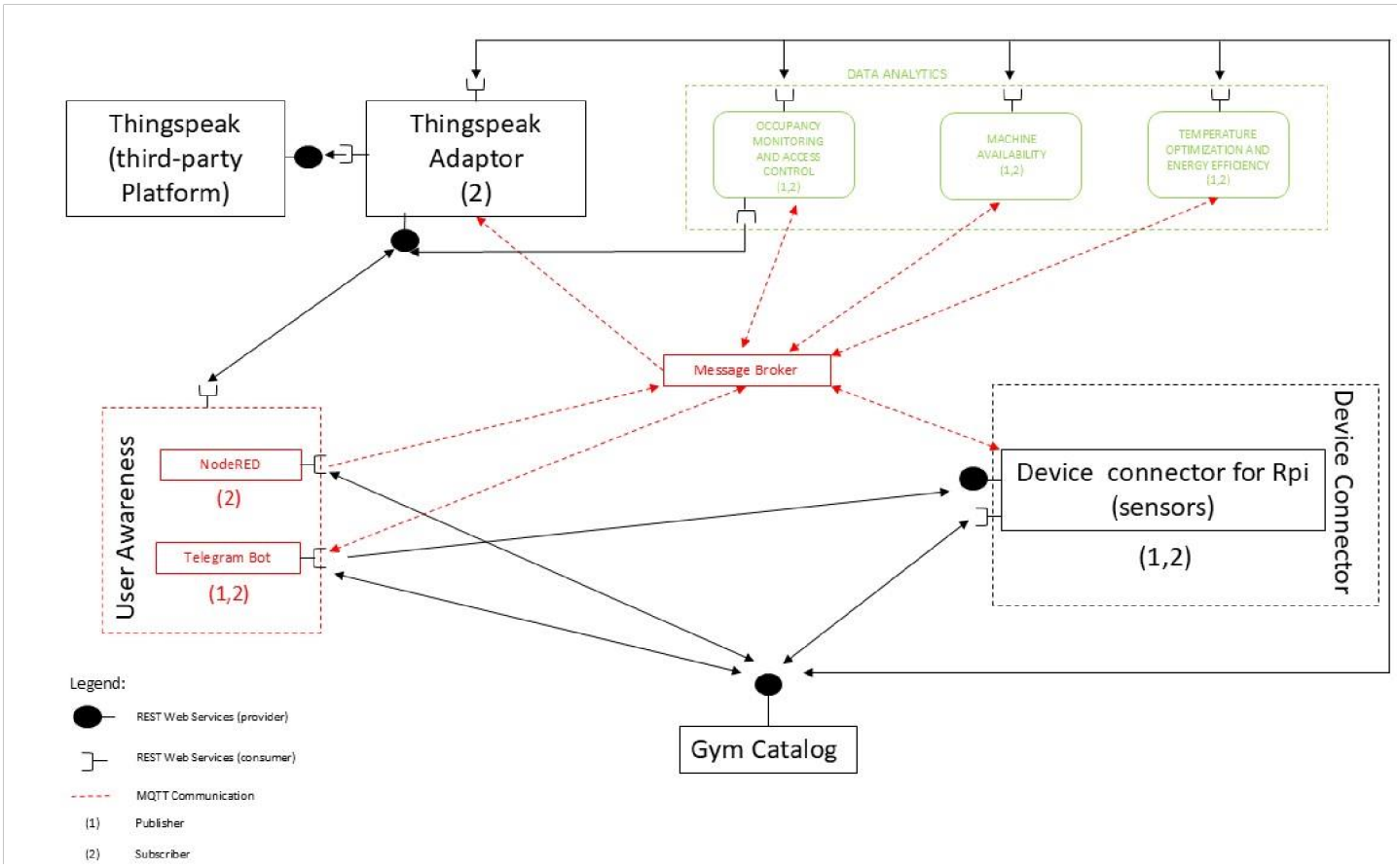
1. Name of Use Case

Name of the Use Case	**Gym Genius**
Version No.	v0.4
Submission Date	19/05/2025
Team Members (with student ids)	Yasmin Awed s320303, Augusta Castelli s320230, Shurui Chen s307957, Federico Villata s319922

2. Scope and Objectives of Function

Scope and Objectives of Use Case	
Scope	The purpose of the proposed IoT platform is to offer services for smart gym management.
Objective(s)	The IoT application Gym Genius is intended to optimize gym management, improve customer satisfaction, and guarantee optimal utilization of resources. It employs automated mechanisms, a variety of sensors, and a Telegram bot to facilitate user interaction.
Domain(s)	Smart Building, Smart Facilities
Stakeholder(s)	Gym owners/managers, gym staff and gym users/members
Short description	The goal of the presented IoT platform is to offer a smart solution to manage a gym environment and optimize customer experience. It incorporates a variety of features such as: • the possibility to check whether a particular type of machine is accessible via an infrared sensor for each machine; • the ability to track the gym's occupancy through a push button that opens the door for individual entry/exit; • the capacity to efficiently control, both remotely or in an automated way, the building's HVAC (Heating, Ventilation, and Air Conditioning), to minimize energy waste and encourage environmentally friendly behavior.

3. Diagram of Use Case



4. Complete description of the system

Gym Genius follows the microservices designing pattern and it makes use of two communication paradigms: request/response based on REST Web Services and publish/subscribe based on MQTT protocol. Ten actors have been identified in this context and presented in the following:

- The **Message Broker** offers a publish/subscribe method-based asynchronous communication exploiting the MQTT protocol.

- The **Gym Catalog** is composed of two parts: the **Service Catalog** and the **Resource Catalog**. The Service Catalog keeps track of all active services, while the Resource Catalog registers all devices within the system. Each service or device automatically registers itself with the appropriate catalog upon initialization. The Gym Catalog provides details on all endpoints (e.g., REST Web Services and MQTT topics) associated with devices, resources, and services on the platform. Additionally, it offers configuration options for control strategies and applications. Services rely on the Service Catalog's REST Web Services to discover and communicate with other services by retrieving their endpoint information.

- The **Device Connector** integrates the sensors with the platform using Raspberry Pi boards. The Raspberry Pi is equipped with temperature and humidity sensors to provide environmental information about the building's condition, infrared sensors to detect the occupancy of specific machines, and push buttons to monitor the entrance and exits of the gym. It offers REST Web Services specifically designed to expose environmental data (e.g., temperature and humidity). Additionally, it utilizes REST to communicate with the Gym Catalog, retrieving information about services and devices registered in the system.
  The Device Connector also functions both as an MQTT publisher and MQTT subscriber. As an MQTT publisher, it transmits data related to the environment (temperature, humidity), occupancy, and machine availability. As an MQTT subscriber, it receives actuation commands to control connected appliances.
  The frequency at which environmental data is published via MQTT is configurable.
  To scale the platform, multiple Raspberry Pis can be deployed in different rooms (such as the entrance, changing rooms, and activity rooms), each equipped with sensors tailored to their specific needs (temperature/humidity sensors, push buttons, infrared sensors, etc.).

- The **Occupancy monitoring and access control** serve as the backbone for monitoring customer presence within the gym, enabling real-time occupancy updates recovered from device connectors. Moreover, the microservice exploits algorithms to make predictions about the occupancy of the gym at different time intervals of the week to optimize gym operations and client experiences. Integration of MQTT and REST facilitates efficient data handling: the service acts as an MQTT subscriber to receive user entry and exit events, and as an MQTT publisher to share current occupancy levels and future predictions. It also implements REST APIs to register itself at startup, obtain configuration and service information from the Gym Catalog, and retrieve historical data from the ThingSpeak Reader.

- The **Machine availability** block tracks the availability of specific gym machine types, showing which ones are being used or not. It works as an MQTT subscriber to receive information about the availability of each individual machine and it acts as an MQTT publisher to transmit Aggregated Machine Availability data, providing an overview of overall machine usage in the gym, which can be used by other services or platforms (eg. Telegram Bot). Additionally, it utilizes RESTful APIs both to register itself at startup and to obtain configuration and service information from the Gym Catalog.

- The **Temperature optimization and energy efficiency** block gathers information from humidity and temperature sensors installed across all areas of the gym. Its goal is to optimize HVAC (heating, ventilation, and air conditioning) usage by considering both environmental conditions and occupancy levels, minimizing energy waste during off-peak hours. The service subscribes to MQTT topics to receive environmental measurements, occupancy status, desired temperature updates, and manual HVAC commands. It publishes MQTT messages to send actuation commands to the Device Connector and to issue alerts when values exceed predefined thresholds. Additionally, it uses REST APIs to register itself at startup and to retrieve configuration and service information from the Gym Catalog.

- The **Thingspeak Adaptor** is a microservice that acts as an MQTT subscriber, listening to topics related to temperature, humidity, occupancy, and machine availability. It uploads the received data to the ThingSpeak platform in real time using REST Web Services. In addition to this real-time data publishing component, the system includes a complementary module called the **ThingSpeak Reader**, which performs the reverse operation. It periodically retrieves historical data from ThingSpeak through HTTP GET requests. The retrieved data is stored locally as structured CSV files for each monitored room and made accessible through a lightweight HTTP interface, allowing external tools to fetch the most recent historical data. Both components use REST APIs to register themselves at startup and to obtain configuration and service information from the Gym Catalog.

- **Thingspeak** is a third-party platform (https://thingspeak.com/) that provides REST Web Services, allowing the ThingSpeak Adaptor to upload real-time data and the ThingSpeak Reader to retrieve historical data. It is an open-data platform for the Internet of Things, designed to store, process, and visualize data through interactive plots.

- **NodeRED** is a dashboard used to retrieve real-time data from the Device Connector via its REST Web Services, import historical data from the ThingSpeak Reader, and exchange information with the Gym Catalog. It also works as an MQTT subscriber to receive additional live data. Node-RED is primarily employed to plot and visualize system metrics through interactive dashboards.

- **Telegram Bot** is a service that connects the Gym Genius infrastructure to the cloud-based instant messaging platform Telegram. It allows users, both clients and administrators, to interact with the system via chat commands and receive real-time updates. It retrieves environmental data by accessing the REST Web Services provided by the Device Connector and uses the Gym Catalog both to register itself at startup and to obtain configuration and service information. Additionally, it functions as both an MQTT subscriber, receiving real-time data such as current occupancy, equipment availability, temperature alerts, and traffic predictions, and an MQTT publisher, sending actuation commands for HVAC control such as on/off switches and mode settings. Through interactive messages, the bot provides access to data visualizations, forecast insights, and administrative control features, making it a key interface for user interaction and system management.

5. Hardware components (only among those we can provide)

No additional devices are needed.