**Application Configuration Report**

Operating System: Raspberry Pi 4

MQTT Integration:

The system leverages the MQTT (Message Queuing Telemetry Transport) protocol for communication.

The Raspberry Pi (embedded system) acts as an MQTT client, connecting to an MQTT broker ('83e374a4d4df41b5926fac056d1f5340.s1.eu.hivemq.cloud').

The 'Deborah\_Security\_System' client subscribes to the 'access\_input' topic to receive messages.

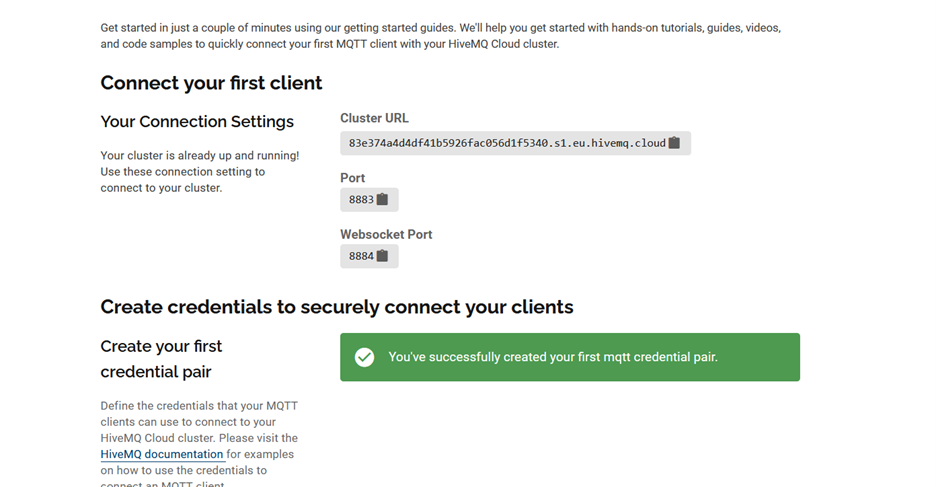


Figure : MQTT Broker Configurations

A screenshot of a computer

Description automatically generated

Figure MQTT Broker Utilization

A screenshot of a computer

Description automatically generated

Figure HiveMQ Broker Interface - Client Messages

Logging Integration:

System events, particularly PIN entries, are logged to a file ('access\_log.txt') using the Python logging module.

Each log entry includes a timestamp and details about the PIN entered.

A screenshot of a computer

Description automatically generated

Figure Security System Log

OLED Display Integration:

The system integrates with an OLED display (SSD1306) to provide visual feedback.

The display is used to prompt users for PIN input and to show messages like "Access Granted" or "Access Denied."

A circuit board with colorful wires

Description automatically generated

Figure PIN Entry

A circuit board with many colored wires

Description automatically generated

Figure Access Granted - Green LED ON

A circuit board with many colored wires

Description automatically generated

Figure Access Denied - Red LED ON

GPIO (General Purpose Input/Output) Integration:

GPIO pins are utilized for interfacing with external hardware components.

Row and column pins are configured to interact with a keypad for PIN entry.

Trigger, green bulb, and red bulb pins are controlled to indicate system status.

Web Interface Integration:

The system incorporates a web interface ('index.html') to provide user interaction.

The interface allows users to log in, view an image, upload files, and access an access log section.

JavaScript scripts ('auth.js' and 'main.js') handle authentication, MQTT connectivity, and dynamic content updates.

A screenshot of a computer

Description automatically generated

Figure Web Interface

External Libraries Integration:

The system utilizes the external MQTT.js library (version 4.2.7) for handling MQTT connections.

The library is sourced securely from 'https://cdnjs.cloudflare.com/ajax/libs/mqtt/4.2.7/mqtt.min.js'.

Rate Limiting Integration:

A rate-limiting mechanism is implemented to mitigate brute-force attacks on PIN inputs.

This integration ensures a secure and controlled method for entering PINs.

Local Storage Integration:

The system employs local storage to store access log information persistently.

This enables the retention of access logs even after system restarts.

Shutdown and Cleanup Integration:

GPIO cleanup is integrated into the system to release configured pins upon shutdown.

The MQTT client is disconnected, and the loop is stopped, ensuring a proper and controlled system exit.

Security System Authentication:

The system has an authentication mechanism to control access to its features.

Users need to enter a valid username and password to access the security system's functionalities.

These integrations collectively contribute to a comprehensive smart security system, combining hardware interactions, communication protocols, web-based interactions, and external libraries to achieve its security and monitoring objectives.