Fish Feeder:
Smart Aquarium Management System

Higher Diploma in Software Engineering
2023.1F

School of Computing and Engineering
National Institute of Business Management
GALLE

NATIONAL INSTITUTE OF BUSINESS MANAGEMENT

HIGHER DIPLOMA IN SOFTWARE ENGINEERING

Internet of Things

GA/HDSE23.1F

SUBMITTED BY

P.G.P. MADUBASHINI	**GAHDSE231F-007**
D.G.S.S. SATHSARA	**GAHDSE231F-008**
K.M.T.D. BANDARA	**GAHDSE231F-009**

SUPERVISED BY

MR. SUPUN ASANGA

Declaration

"We certify that this project does not, to the best of our knowledge and belief, contain any work that has already been published or authored by someone else or us or that has previously been submitted for a Higher Diploma at any institution. We hereby declare that the work presented in this project report was done independently by us and that we have properly cited the work of others."

P.G.P. MADUBASHINI
D.G.S.S. SATHSARA
K.M.T.D. BANDARA

Abstract

The Fish Feeder project is aimed at automating feeding of pet fish to cater for a number of common problems faced by fish owners such as irregularity in feeding and remote management of the feeding process. It is made up of a NodeMCU ESP8266 microcontroller, Continuous Rotation SG90 Servo Motor, OLED Display, and a Membrane Switch Keypad all integrated with a sturdy software stack that includes Node.js server, React Native mobile app, Firebase, and HiveMQTT broker.

Users can set flexible feeding schedules, initiate manual feeds or monitor their history through the mobile app. Periodically, The Node.js server checks for scheduled feeding tasks and instructs the NodeMCU to dispense food at the appropriate times. The system has real-time notification support as well as data synchronization hence ensuring fish are fed reliably and consistently.

By meticulously integrating hardware-software and extensive testing various functionalities such as real-time communication and hardware compatibility were implemented. Some future improvement may be smart ways of giving food to fish using sensor integration voice control capabilities while others include features like fish health monitoring and integration with smart home systems.

This project successfully demonstrates a comprehensive solution for automated fish feeding, offering enhanced control, convenience, and reliability for pet fish owners.

Acknowledgment

Our Fish Feeder project was completed successfully, and we would like to take this opportunity to express our sincere gratitude towards the contributors.

To begin with, we wish to extend our deepest gratitude to Mr. Supun – our lecturer for the Internet of Things module at NIBM. We are grateful for his guidance, motivation and support that made it possible for us to achieve our goals. Besides, his know-how and ideas laid a firm base for us.

Also, we are thankful to NIBM for providing us with all necessary resources and facilities needed in developing this project. Our success depended on collaborative environment as well as exposure to technology.

Additionally, we value the hard work and commitment from group members. All team players' dedication contributed much on how they did their parts in relation to successful execution of this project. The evidence of collaboration spirit during the entire period enabled us move forward beyond difficulties thereby leading into development of an inclusive approach that resolved everything.

Finally, let's appreciate our families and friends who have been supporting us through thick and thin in this undertaking.

This marks the end of my statement with many thanks again for your contributions and support.

Yours faithfully,
D.G.S.S. Sathsara.
P.G.P. Madubashini.
K.M.T.D. Bandara.

Table of Contents

Table of Figures

1 Introduction

1.1 Background

Pet fish keeping is a common hobby which gives the pleasure and relaxation. Although the visits of the human beings for feeding the fish is not important, still, the continuous feeding schedule is the main factor for the fish to be healthy and happy. People who own fish usually must deal with their hectic schedules, forgetfulness and the need to travel which in turn result in the irregular feeding patterns of the fish. The irregular feeding is the cause of stress, the health problems, and even the death of the fish. Thus, there is a big need for an automatic solution that takes care of the fish's feeding, whether the owner is there or not.

Existing solutions and their limitations:

Many of the automated fish feeders are present on the market providing basic features such as the timed feeding. These gadgets usually let users to set when they should eat and how much they should eat. However, most existing solutions have limitations, including:

- **Limited Scheduling Options:** Most of the commercial feeders provide only the simple scheduling options, usually the scheduling is for a fixed time of the day without the flexibility to set the multiple feeding schedules with different frequencies.
- **Lack of Remote Control:** There are very few devices that have the remote control feature, thus, the feeding schedule has to be changed manually.
- **Absence of Integration with Modern Technology:** A lot of feeders do not work together with smartphones or the internet, thus, putting the user in a tight spot regarding the control and convenience.
- **Inadequate Monitoring:** The current solutions usually do not have the facilities for monitoring the feeding history and fish tank conditions that thus, the users find it difficult to track the feeding patterns and to ensure the best care of the fish.

These constraints show the necessity for a modern and more advanced automated fish feeding system that makes use of technology to offer better control, convenience, and reliability.

1.2 Objectives

The goals and objectives of the Fish Feeder project are as follows:

- **Automate Fish Feeding:** Come up with a system that feeds the fish at the scheduled times, thus making sure that the feeding patterns are consistent and reliable.

- **Remote Control and Monitoring:** Let the users to see and manage the feeding schedule from wherever they are via a mobile app.
- **Flexible Scheduling:** Users will be able to create several feeding schedules with a specific time and day of the week, and the toggling of them on or off will be very easy.
- **User-Friendly Interface:** Design the mobile app and the hardware device with a simple and interactive interface to enhance the user's experience and to make sure that they can easily control it.
- **Real-Time Notifications:** Let the users be notified via real-time about the feeding activities and schedule changes.
- **Data Logging:** Record and show the feeding history in the form of graphs and charts to assist the users in tracking their fish's feeding patterns.
- **Manual Feeding Option:** Add an option for instant manual feeding through the mobile app and the hardware device.

1.3 Scope

The scope of the Fish Feeder project includes the following:

- **Hardware Components:**
    - NodeMCU ESP8266 WiFi module for connectivity and control.

Figure 1 NodeMCU ESP8266 WiFi module

- Continuous Rotation SG90 Servo Motor for dispensing food.

Figure 2 SG90 Servo Motor

- 0.96-inch 128X64 OLED Display for displaying messages and schedules.

Figure 3 OLED Display

- 1x4 4-Key Membrane Switch Keypad for manual control.

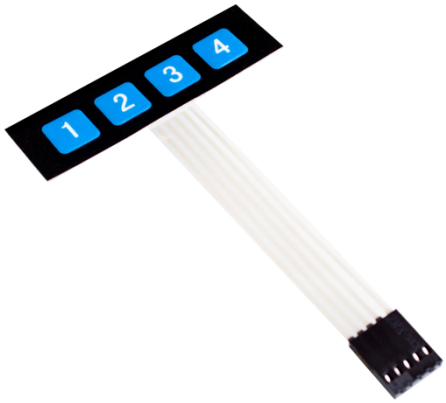

Figure 4 4-Key Membrane Switch Keypad

- **Software Components:**
  - A Node.js server to handle scheduling logic and to communicate with the hardware.
  - A React Native mobile app for UI and control.
  - Firebase as the database for storing user schedules and feeding history.
  - HiveMQTT broker for message communication between the server and NodeMCU.
- **Functionalities:**
  - The system of feeding automatically depending on the user-defined time schedule.
  - Manual feeding via mobile app and hardware keypad.
  - Display of current schedules and status on the OLED screen.
  - Remote control and schedule management through the mobile app.
  - Real-time notifications and feeding histograms.
- Limitations:
  - The reliance on a stable internet connection for the remote-control features.
  - Initial setup is complex for user who don’t have any experience with IOT devices.
  - The probable hardware restrictions concerning the storage of food and the motor life.

2 Methodology

2.1 System Architecture

The Fish Feeder is a system that is created to automate the feeding process for pet fish by the means of the hardware and software elements. The system comprises three main components: the gear: NodeMCU ESP8266, Servo Motor, OLED Display, Keypad, the software: NodeJs server, React Native app, Firebase, HiveMQTT broker, and the network infrastructure.

Components and interactions:

- **NodeMCU ESP8266:** Being the main controller and the one who gets the commands from the Node and controls Servo Motor and OLED Display.
- **Continuous Rotation SG90 Servo Motor:** responsible for the dispensing of fish food based on the commands from the NodeMCU.
- **OLED Display:** Gives a user interface on the hardware side that shows the messages and schedule information.
- **Membrane Switch Keypad:** enables the users to manually interact with the system, enabling them to feed immediately, schedule, and toggle schedules.
- **Node.js Server:** takes care of the scheduling logic, handles the requests of the users, and talks to Firebase and the NodeMCU.
- **React Native App:** through this user are able to control and monitor the fish. They can set the schedules, feed the fish manually, and have a history of feeding events.
- **Firebase:** Serves as the database, storing user schedules and feeding history.
- **HiveMQTT Broker:** Act as a bridge to communicate between NodeJS server and NodeMCU.

2.2 Hardware Components

- **NodeMCU ESP8266:**
    - A low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability.
    - Features: Built-in Wi-Fi, GPIO pins for interfacing with other hardware components, programmable via Arduino IDE.
- **Continuous Rotation SG90 Servo Motor**:
    - A small, lightweight servo motor that can rotate continuously in either direction.
    - Used to dispense fish food by rotating a food container or mechanism.
- **0.96-inch 128X64 OLED Display:**

- A small, energy-efficient display module with a resolution of 128x64 pixels.
- Used to display messages, schedules, and status updates to the user.

- **1x4 4-Key Membrane Switch Keypad:**
  - A simple keypad with four buttons labeled 1 to 4.
  - Functions:
    - Key 1: Instant feeding.
    - Key 2: Check schedules and scroll through them.
    - Key 3: Toggle active status of schedules.
    - Key 4: Return to Home Screen.

2.3 Software Components

- Node.js Server:
  - Backend logic and scheduling.
  - Interfaced with Firebase to fetch and update schedule data.
  - It employs HiveMQTT for feeding commands publishing to the NodeMCU.
- React Native App:
  - A cross-platform mobile interface for users.
  - With home screen for instant feeding, schedule screen for managing feeding times, history screen for viewing feeding logs.
  - talks with Node.js server by means of REST API.
- Firebase:
  - A cloud-based database where user's data such as feeding schedules and history can be saved.
  - P It enables real-time data synchronization as well as storage provision.
- HiveMQTT Broker:
  - A messaging protocol for small sensors and mobile devices, optimized for high-latency or unreliable networks.
  - This acts an intermediary between the Node.js server and the NodeMCU for real-time communication facilitation.

2.4 Data Flow

1. User Interaction:

- The user interacts with the mobile app to set feeding schedules or initiate an instant feed.
- The app sends requests to the Node.js server to update schedules or trigger feeding.

2. Server Processing:
    - The Node.js server processes the requests and updates the Firebase database accordingly.
    - For scheduled feeding, the server periodically checks Firebase for upcoming feeding tasks.
3. Command Transmission:
    - When it's time to feed the fish, the Node.js server publishes a feeding command to the HiveMQTT broker.
    - The NodeMCU, subscribed to the MQTT topic, receives the command.
4. NodeMCU Action:
    - The NodeMCU activates the Servo Motor to dispense food.
    - Same time, it updates the OLED Display with a feeding message.
5. Data Feedback:
    - After feeding, the NodeMCU sends a confirmation back to the server.

3 Discussion

3.1 Implementation Details

- Automated Feeding:
    - Node.js server, Firebase database, and NodeMCU hardware combined were used to implement the automated feeding functionality.
    - The Firebase database is periodically checked by the Node.js server for scheduled feeding tasks.
    - When the scheduled feeding task is found, server sends a command to HiveMQTT broker and then the NodeMCU to start food dispensing process.
    - A feeding message appears on OLED Display as the Servo Motor is controlled by NodeMCU.
- Remote Control and Monitoring:
    - The React Native mobile app was used to implement remote control and monitoring functionality.
    - The intuitive interface of this app allows users to set up feeding schedules, initiate feeds manually or view their feed history.
    - Scheduling updates and feeding commands are communicated to the Node.js server through REST API by this app.
    - Firebase based real-time data synchronization ensures updated information about fish feeding activities for users.
- Flexible Scheduling:
    - It allows one user to have different times and days in a week for their multiple schedules of fish feeding services.
    - Schedules can be turned ON/OFF using either a mobile phone application or a hardware keypad as they are stored in Firebase.
    - On all these issues, scheduling logic is handled by the Node.js server such that proper time management is ensured while providing food services.

3.2 Challenges and Solutions

- Hardware Integration:
    - Assembling the wired and compatible parts like NodeMCU, Servo Motor, OLED Display and Keypad among others was challenging.

    - Solution: A thorough experiment was done to ensure that the hardware is properly integrated. The process became easy when documented clearly with pin mappings.
- Real-Time Communication:
    - There are problems in message delivery and synchronization since one main issue is ensuring real-time communication between mobile app, node.js server, NodeMCU.
    - Solution: The process involved the use of MQTT messaging protocol as well as Firebase real-time database which provided reliable communication channels. Besides dealing with connectivity issues by way of error handling mechanisms.
- Testing and Validation:
    - Extensive testing had to be conducted for this system to confirm its functionality under various scenarios such as different network conditions or scheduling configurations etc.
    - Solution: For this purpose, exhaustive testing protocols were developed including unit tests, integration tests and end-to-end tests. Emulators as well as physical testing environments were employed to mimic real-world situations.

3.3 Testing and Validation

Methods used to test the system:

- **Unit Testing:** This involves testing of individual components such as the Node.js server, React Native app, and NodeMCU firmware to ensure they operate according to design.
- **Integration Testing:** It tests the integration of hardware and software components for compatibility checks and communication barriers.
- **End-to-End Testing:** The whole system was tested on a simulated environment to assess its performance under normal conditions experienced in real world.
- **User Acceptance Testing:** Evaluating the system by potential users with an aim of receiving feedback from them and also identifying any problems concerning usability.

Results of testing:

- The mechanism illustrated dependable automated feeding functionality whereby feeding tasks were carried out precisely as set up by users' schedules.
- Remote control along monitoring capabilities offered intuitive interfaces that allowed user to manage their feeds schedule as well as monitor past feeds.

- During testing, the system proved itself robust and stable with minimal downtimes or errors that could be observed.
- User’s feedback shows high satisfaction levels for both usability and performance of this system further indicating that it met project objectives effectively.

4 Future Implementations

4.1 Enhancements

- **Smart Feeding Algorithms:** Smart feeding algorithms should be established and they must be able to take into account the species of fish, their sizes as well as feeding behaviors thus optimizing how much and how often these are fed.
- **Sensor Integration:** Sensor integration will enable monitoring of water parameters such as pH, temperature, ammonia levels among others thus enabling real-time monitoring and automated adjustments of the feeding schedules.
- **Voice Control:** This can also be achieved by adding voice control functionalities on it which will allow people to speak with the feeder using human language for a handsfree operation.
- **Machine Learning Integration:** Machine learning algorithms can be used to analyze data and provide personalized feeding schedules and quantities.
- **Enhanced Notification System:** Can improve the notification system by adding more notifications methods such as SMS and emails.

4.2 Additional Features

- **Fish Health Monitoring:** The health monitoring elements should be merged with function to observe fish actions and detect the abnormalities that will be the signs of the problems with the health, like the unusual swimming mannerisms or the lack of appetite.
- **Automatic Food Dispensing:** Develop an automatic food dispensing system that gives the exact amounts of food at the pre-determined feeding times thus, it reduces the food waste and keeps the food quantities unified.
- **Integration with Smart Home Systems:** The fish feeder system can be connected to gadgets used for smart homes like Google Home or Amazon Alexa that enable the users to manage and watch feeding activities through voice commands and also, schedule smart home automation routines.
- **Social Sharing:** The users can exchange their feeding schedules and histories with other people who are into the same hobby or they can be a part of online communities thus, promoting the interaction within the community and the knowledge sharing among the people.

- **Augmented Reality (AR) Feeding Experience:** Design a virtual reality where the users have the virtual fish tanks in which they can feed them and control the feeding schedules at any time by means of the real-time interactions.
- **Environmental Enrichment Features:** The environmental enrichment aspects like interactive toys and the simulated natural habitat can be put in order to increase both the welfare and the stimulation which comes with keeping pet fish.

5 References

- learn.sparkfun.com. (n.d.). ESP8266 Thing Hookup Guide - SparkFun Learn. [online] Available at: https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide.
- Firebase (2019). *Firebase Realtime Database*. [online] Firebase. Available at: https://firebase.google.com/docs/database.
- www.hivemq.com. (n.d.). *MQTT Essentials - All Core Concepts explained*. [online] Available at: https://www.hivemq.com/mqtt-essentials/.
- Anon, (2019). *ESP32 OLED Display with Arduino IDE | Random Nerd Tutorials*. [online] Available at: https://randomnerdtutorials.com/esp32-ssd1306-oled-display-arduino-ide/.
- Space Docs. (n.d.). *Run a Node.js App 🚀 Space Docs*. [online] Available at: https://deta.space/docs/en/build/quick-starts/node/ [Accessed 15 May 2024].

6 Gantt Chart

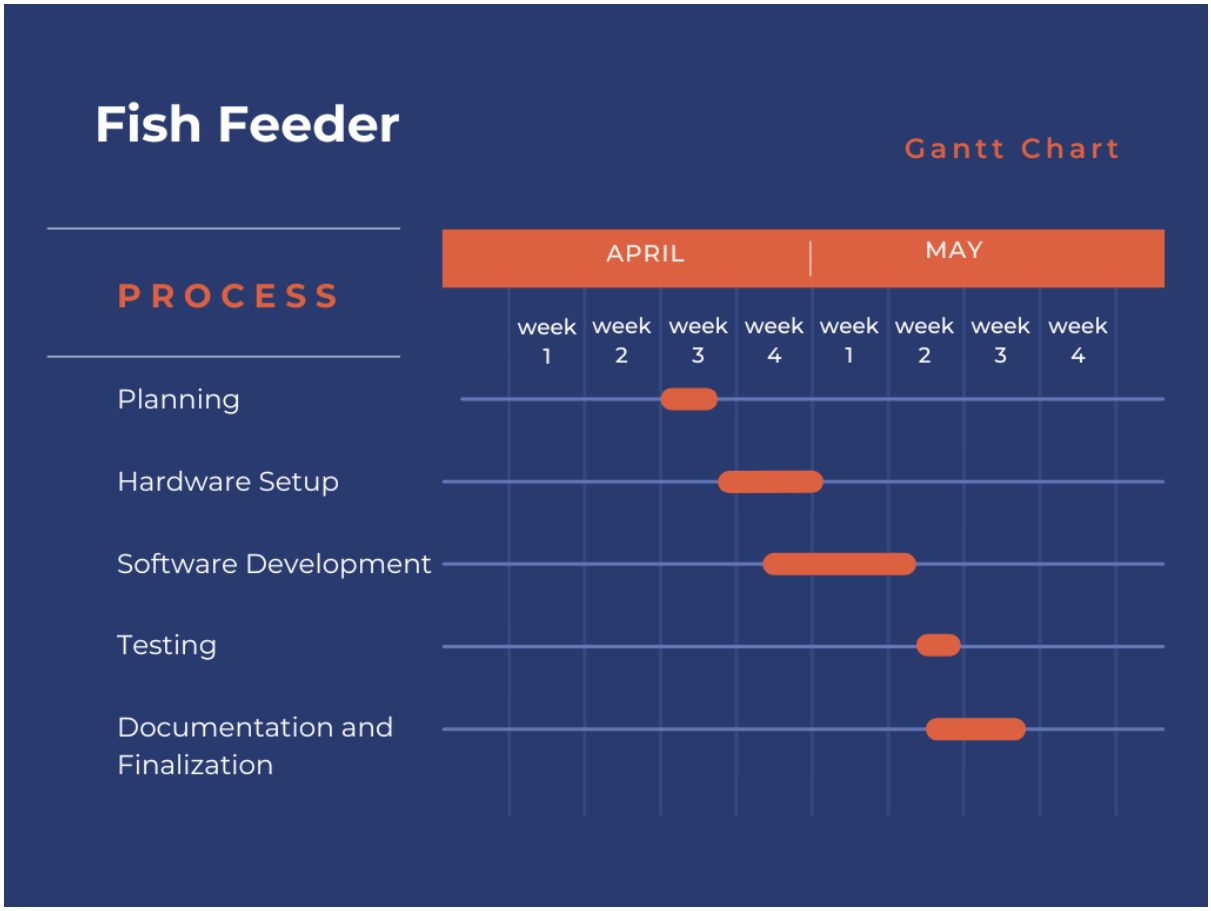

Figure 5 Gantt Chart

7 Appendices

7.1 7.1 Screenshots

Figure 6 Display Home screen

Figure 7 Feeding Message

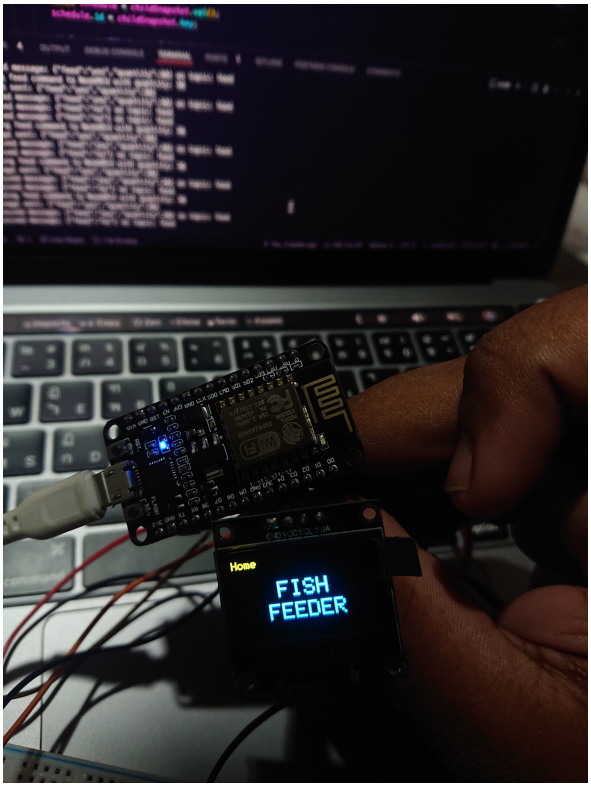

Figure 8 NodeMCU Module

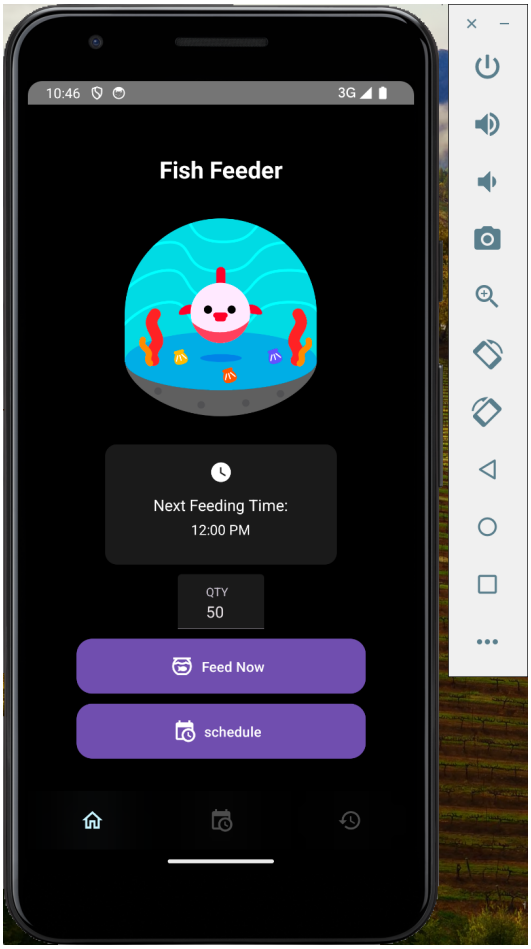

Figure 9 App Home Screen

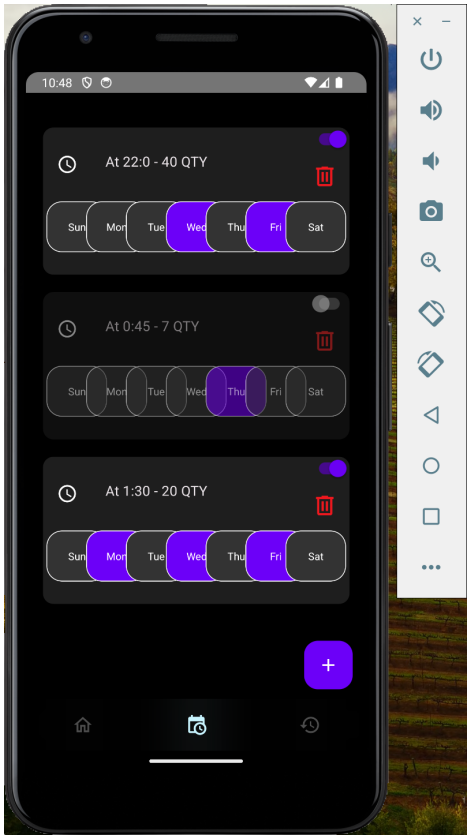

Figure 10 App: Manage schedules

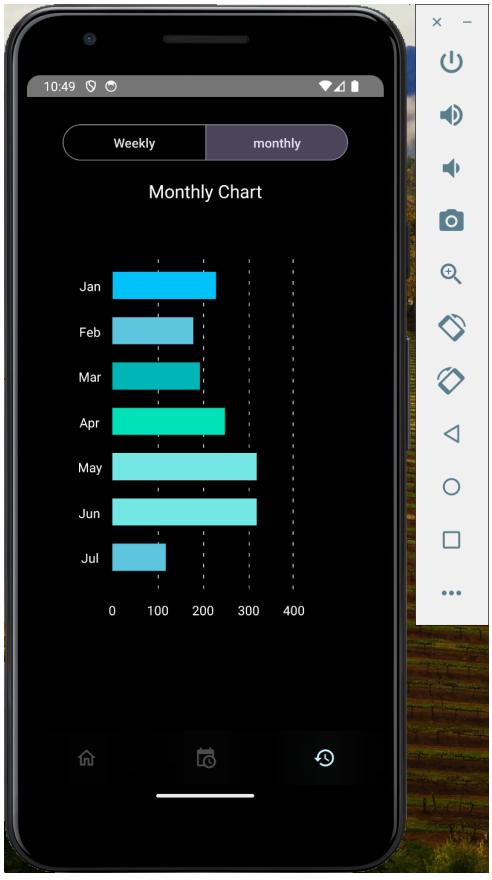

Figure 11 App: Monthly Histogram

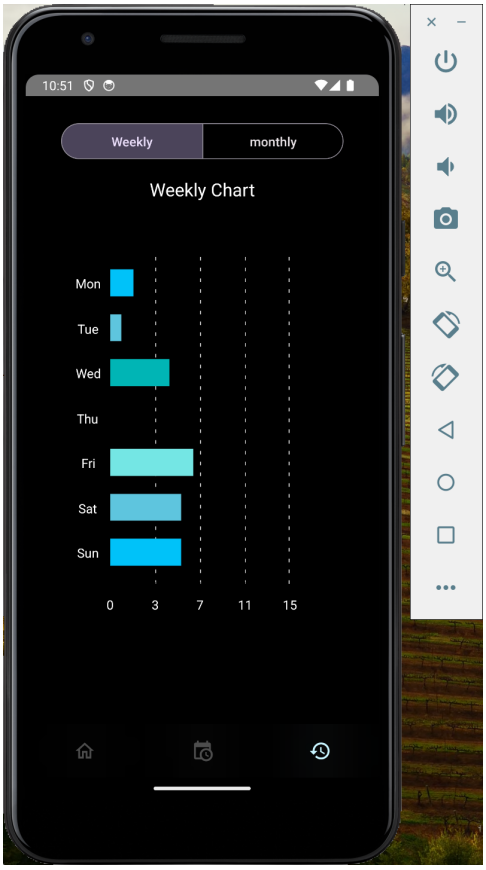

Figure 12 App: Weekly Histogram

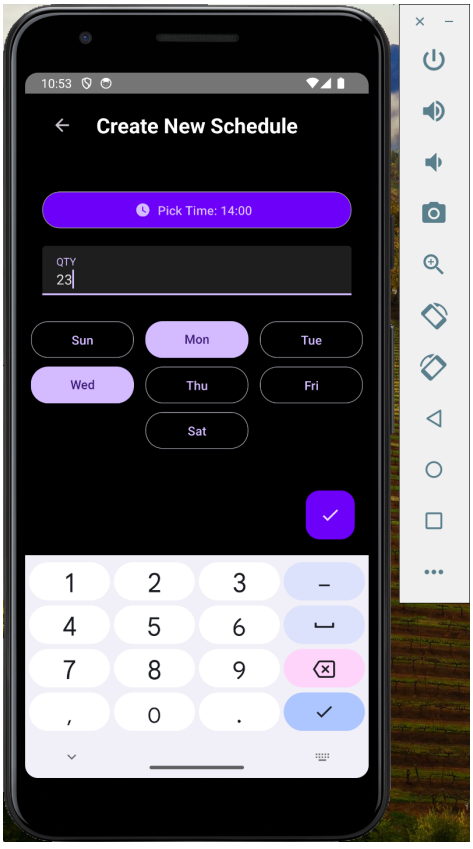

Figure 13 App: Create schedules

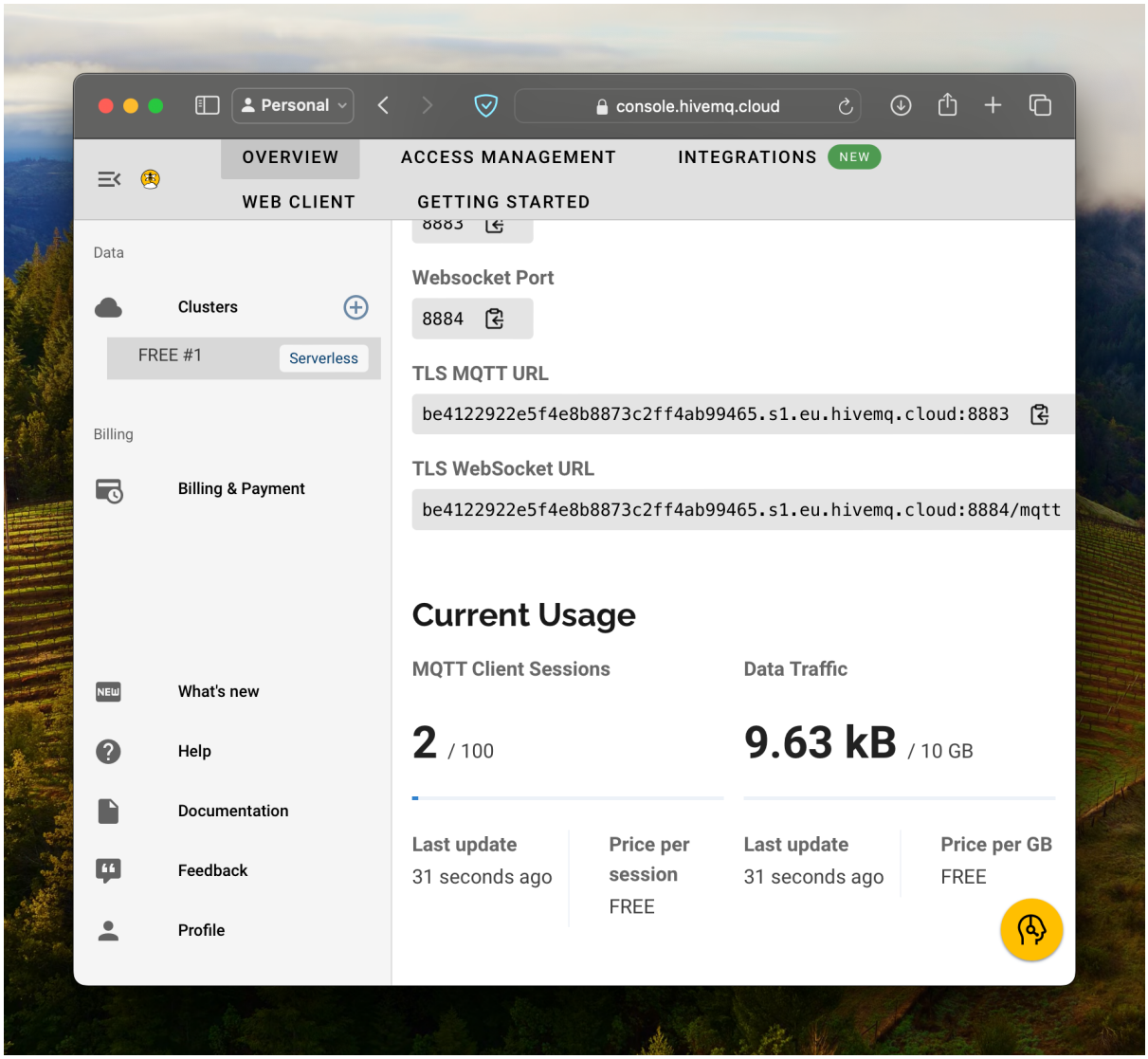

Figure 14 MQTT broker dashboard