Software Requirements Specification

For

<Bể cá thông minh>

Phiên bản 1.0 đã được phê duyệt

Người soạn thảo: Nguyễn Văn Lý

Vũ Đăng Trường Giang

Nhóm : 3

Ngày soạn thảo : 14/01/2025

**MỤC LỤC**

**MỤC LỤC i**

**LỊCH SỬ CHỈNH SỬA ii**

[**1. Giới thiệu**](#_c46ex08b7nf2) **1**

[1.1 Mục đích](#_ec1gimok0rhk) 1

[1.2 Quy ước tài liệu](#_ukb7j3ra373) 1

[1.3 Đối tượng người đọc và gợi ý đọc](#_90ru809k7bku) 1

[1.4 Phạm vi dự án](#_mxzjatwfsclc) 2

[1.5 Tài liệu tham khảo](#_cmz6w2d5rtkj) 2

[**2. Mô tả tổng quan**](#_mx49fcng9s31) **2**

[2.1 Quan điểm về sản phẩm](#_ei059t40k2z8) 2

[2.2 Tính năng của sản phẩm](#_slzxbjnjrbn6) 2

[2.3 Các lớp người dùng và đặc điểm của họ](#_ax0eh3cydz6q) 2

[2.4 Môi trường hoạt động](#_h6gxped5cs5j) 2

[2.5 Các ràng buộc thiết kế và triển khai](#_8fckjft2almq) 3

[2.6 Tài liệu người dùng](#_uj60i64op7oh) 3

[2.7 Các giả định và phụ thuộc](#_jo5hlji9fuba) 3

[**3. Tính năng hệ thống**](#_4hr4b36ddnom) **3**

[3.1 Hiển thị dữ liệu cảm biến](#_1m6ytvws2x34) 3

[3.2 Cảnh báo và thông báo](#_kengyfycsloq) 3

[3.3 Lịch sử dữ liệu](#_7vyuep7l3dtb) 3

[**4. Yêu cầu giao diện ngoài**](#_mkp8gv9f7x2q) **4**

[4.1 Giao diện người dùng](#_53k00g1kyi00) 4

[4.2 Giao diện phần cứng](#_fepvyd387exv) 4

[**5. Các yêu cầu phi chức năng khác**](#_rkmzuurlb1gz) **4**

[5.1 Yêu cầu về hiệu suất](#_h1y7cfgyiqdm) 4

[5.2 Yêu cầu về an toàn](#_g5yen0etg8pf) 4

[5.3 Yêu cầu về bảo mật](#_pg3gxffjg66) 4

[5.4 Các đặc tính chất lượng phần mềm](#_j16id3mr8ppd) 5

[**6. Các yêu cầu khác, các cải tiến có thể có để giúp sản phẩm phát triển**](#_a788ehyd5orx) **5**

[**Phụ lục A: Từ vựng**](#_yszccwobbb3i) **5**

[**Phụ lục B: Mô hình phân tích**](#_u6csa56r0rwh) **5**

[**Phụ lục C: Danh sách vấn đề**](#_fuoitx2dh0vl) **5**

Lịch sử chỉnh sửa

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Ngày** | **Lý do thay đổi** | **Phiên bản** |
| Bể cá thông minh | 14/01/2005 |  | 1.0 |
|  |  |  |  |

### **1. Giới thiệu**

#### **1.1 Mục đích**

Tài liệu này mô tả các yêu cầu cho hệ thống bể cá thông minh, nơi các cảm biến như nhiệt độ nước, pH, chất lượng nước sẽ gửi dữ liệu lên trang web local host, cho phép người dùng giám sát và có thể điều khiển các thông số yếu tố trong bể cá.

#### **1.2 Quy ước tài liệu**

Tài liệu này xác định tuân theo định dạng chuẩn SRS chuẩn. Các thuật ngữ quan trọng được in nghiêng, các yêu cầu hệ thống được đánh số tuần tự, đảm bảo tính minh bạc

#### **1.3 Đối tượng người đọc và gợi ý đọc**

Tài liệu này dành cho các nhà phát triển phần mềm, các kỹ sư phần cứng, và những người tham gia vào việc thiết kế, phát triển và triển khai hệ thống bể cá thông minh.

#### **1.4 Phạm vi dự án**

Dự án bao gồm các thành phần chính:

* Các cảm biến đo nhiệt độ nước, độ pH, và chất lượng nước.
* Một hệ thống web local host để hiển thị thông tin từ các cảm biến.
* Một hệ thống cảnh báo thông qua giao diện web khi có sự cố.
* Sử dụng ESP32 làm vật trung gian truyền tải dữ liệu của cảm biến để hiển thị lên web.

#### **1.5 Tài liệu tham khảo**

* Các tài liệu về giao thức HTTP và WebSocket.
* Tài liệu hướng dẫn sử dụng phần cứng cảm biến (ví dụ, cảm biến nhiệt độ, cảm biến pH).

### **2. Mô tả tổng quan**

#### **2.1 Quan điểm về sản phẩm**

Sản phẩm sẽ bao gồm phần cứng cảm biến kết nối với máy chủ localhost thông qua ESP32, nơi dữ liệu từ các cảm biến được hiển thị trên giao diện web. Người dùng có thể giám sát các thông số môi trường của bể cá, bao gồm nhiệt độ, pH, mức oxy trong nước thông qua một trang web nội bộ.

#### **2.2 Tính năng của sản phẩm**

* **Hiển thị dữ liệu cảm biến**: Dữ liệu từ các cảm biến sẽ được cập nhật và hiển thị trên giao diện web local host theo thời gian thực.
* **Cảnh báo và thông báo**: Khi có sự cố về nhiệt độ, độ pH hoặc chất lượng nước không đạt yêu cầu, hệ thống sẽ gửi cảnh báo.

#### **2.3 Các lớp người dùng và đặc điểm của họ**

* **Chủ bể cá**: Người dùng chính sẽ giám sát và theo dõi các thông số của bể cá qua trang web local host.
* **Người quản lý hệ thống**: Các kỹ thuật viên hoặc người bảo trì sẽ hỗ trợ duy trì hệ thống phần cứng và phần mềm.

#### **2.4 Môi trường hoạt động**

* ESP32 kết nối các cảm biến và truyền dữ liệu tới máy chủ web chạy trên local host (có thể là một máy tính hoặc Raspberry Pi).
* Các cảm biến được kết nối với ESP32 qua giao tiếp UART, I2C hoặc SPI.
* Giao diện người dùng web sẽ hiển thị dữ liệu cảm biến qua giao thức HTTP hoặc WebSocket.

#### **2.5 Các ràng buộc thiết kế và triển khai**

* ESP32 phải có khả năng tiếp nhận và truyền dữ liệu của cảm biến theo thời gian thực.
* Giao diện web phải dễ sử dụng và phản hồi nhanh khi có thay đổi trong các giá trị cảm biến..

#### **2.6 Tài liệu người dùng**

* Hướng dẫn sử dụng trang web để theo dõi dữ liệu cảm biến.
* Hướng dẫn cấu hình hệ thống để hiển thị dữ liệu cảm biến.

#### **2.7 Các giả định và phụ thuộc**

* Hệ thống cần có một mạng nội bộ ổn định để đảm bảo kết nối giữa cảm biến và máy chủ localhost.
* Các cảm biến có khả năng cung cấp dữ liệu liên tục và chính xác.

### **3. Tính năng hệ thống**

#### **3.1 Hiển thị dữ liệu cảm biến**

* **Mô tả**: Dữ liệu từ các cảm biến nhiệt độ, pH, và chất lượng nước sẽ được truyền về máy chủ và hiển thị trên giao diện web.
* **Yêu cầu**:
  + Các giá trị từ cảm biến sẽ được cập nhật và hiển thị theo thời gian thực.
  + Giao diện web cần có các biểu đồ hoặc bảng để người dùng dễ dàng theo dõi dữ liệu.

#### **3.2 Cảnh báo và thông báo**

* **Mô tả**: Hệ thống sẽ tự động gửi cảnh báo nếu các thông số môi trường vượt quá ngưỡng cho phép.
* **Yêu cầu**:
  + Cảnh báo phải xuất hiện trên giao diện web cho người dùng.
  + Các ngưỡng cảnh báo có thể được người dùng cấu hình qua trang web.

#### **3.3 Lịch sử dữ liệu**

* **Mô tả**: Giao diện web cần có khả năng lọc dữ liệu theo thời gian.

**3.4 Giao tiếp với ESP32**

Mô tả: ESP32 sẽ được sử dụng làm trung gian truyền tải dữ liệu từ các cảm biến đến máy chủ localhost.

Yêu cầu:

* ESP32 phải có khả năng thu thập dữ liệu từ các cảm biến thông qua các giao thức như I2C, UART, hoặc GPIO.
* Dữ liệu từ ESP32 sẽ được gửi đến máy chủ thông qua giao thức HTTP, WebSocket hoặc MQTT.
* ESP32 phải có khả năng xử lý và gửi dữ liệu theo thời gian thực, với độ trễ tối đa là 1 giây.
* ESP32 phải hỗ trợ kết nối Wifi ổn định để đảm bảo truyền tải dữ liệu liên tục.
* Trong trường hợp mất kết nối, ESP32 cần lưu trữ tạm thời dữ liệu và gửi lại khi kết nối được khôi phục.

### **4. Yêu cầu giao diện ngoài**

#### **4.1 Giao diện người dùng**

* Giao diện web phải hiển thị các giá trị từ các cảm biến (nhiệt độ, pH, oxy hòa tan, độ cứng).
* Giao diện cần có khả năng hiển thị biểu đồ thời gian thực, cảnh báo và thông báo cho người dùng.

#### **4.2 Giao diện phần cứng**

* Các cảm biến sẽ kết nối với máy chủ qua **giao thức HTTP**, **WebSocket** hoặc **MQTT**.
* Dữ liệu cảm biến sẽ được gửi đến máy chủ local host và hiển thị trên giao diện web.

**4.3 Giao diện ESP32 với cảm biến**

* ESP32 cần hỗ trợ tích hợp với các cảm biến được sử dụng (nhiệt độ, pH, oxy hòa tan, độ cứng) thông qua các giao thức giao tiếp phù hợp.
* ESP32 cần có khả năng cấu hình ngưỡng cảnh báo trên các cảm biến và gửi dữ liệu đã qua xử lý về máy chủ.
* Cần triển khai mã nguồn ESP32 với các chức năng tự kiểm tra kết nối và tự khởi động lại trong trường hợp gặp sự cố.

### **5. Các yêu cầu phi chức năng khác**

#### **5.1 Yêu cầu về hiệu suất**

* Hệ thống phải hiển thị dữ liệu cảm biến lên giao diện web trong vòng 2 giây sau khi nhận dữ liệu từ cảm biến.
* Giao diện web phải có thể xử lý và hiển thị dữ liệu liên tục mà không gặp vấn đề về độ trễ.

#### **5.2 Yêu cầu về an toàn**

* Dữ liệu giữa các cảm biến và máy chủ web phải được mã hóa để đảm bảo bảo mật khi truyền tải.

#### **5.3 Yêu cầu về bảo mật**

* Hệ thống web phải được kết nối vào local host riêng để người dùng có thể truy cập và thay đổi các thiết lập.

#### **5.4 Các đặc tính chất lượng phần mềm**

* Giao diện web phải dễ sử dụng, dễ hiểu và trực quan.
* Các lỗi hệ thống cần được thông báo rõ ràng và có khả năng xử lý nhanh chóng.

**5.5 Yêu cầu về ESP32**

* ESP32 cần được lập trình để đảm bảo hiệu suất truyền tải dữ liệu ổn định trong thời gian dài.
* Phần mềm ESP32 cần tối ưu để tiết kiệm năng lượng, đặc biệt nếu hoạt động với nguồn pin.
* ESP32 cần có khả năng cập nhật phần mềm (OTA - Over-the-Air) để dễ dàng thêm tính năng hoặc vá lỗi.

### **6. Các yêu cầu khác, các cải tiến có thể có để giúp sản phẩm phát triển**

* Hệ thống phải có khả năng mở rộng, dễ dàng thêm các cảm biến hoặc tính năng mới mà không ảnh hưởng đến hệ thống hiện tại.
* Giao diện web cần có khả năng hiển thị đa dạng các loại cảm biến khác nhau nếu được thêm vào trong tương lai.

### **Phụ lục A: Từ vựng**

* **WebSocket**: Giao thức mạng cho phép hai chiều giữa máy chủ và trình duyệt, phù hợp với các ứng dụng thời gian thực.
* **HTTP**: Giao thức truyền tải siêu văn bản, cho phép máy chủ và trình duyệt trao đổi dữ liệu.

### **Phụ lục B: Mô hình phân tích**

* Mô hình hệ thống gồm máy chủ localhost, các cảm biến và giao diện web.

### **Phụ lục C: Danh sách vấn đề**

* Các vấn đề có thể gặp phải như lỗi kết nối, độ chính xác của cảm biến hoặc các vấn đề về giao diện người dùng.