Software Requirements Specification

for

Hệ thống dự báo

lượng mưa

Version 1.0 approved

Prepared by Nhóm 9

Organization: Trường Đại Học Điện Lực

Date created:21/2/2024

**Mục lục**

Mục lục ii

Lịch sử chỉnh sửa ii

1. Giới thiệu 1

1.1 Mục đích 1

1.2 Quy ước 1

1.3 Đối tượng và hướng dẫn đọc 1

1.4 Phạm vi dự án 2

1.5 Tài liệu tham khảo 2

2. Mô tả tổng quan 2

2.1 Bối cảnh 2

2.2 Tính năng 3

2.3 Lớp người dùng 3

2.4 Môi trường hoạt động 4

2.5 Hạn chế thiết kế và triển khai 4

2.6 Tài liệu người dùng 5

2.7 Giả định và phụ thuộc 5

3. Tính năng hệ thống 6

3.1 Thu thập và lưu trữ dữ liệu 6

3.2 Nội dung hệ thống hiển thị 6

3.3 Lịch sử và phân tích dữ liệu (Tính năng mở rộng) 6

3.4 Điều khiển thiết bị từ xa (Tính năng mở rộng) 7

4. Yêu cầu giao diện 7

4.1 Giao diện người dùng 7

4.2 Giao diện phần cứng 8

4.3 Giao diện phần mền 9

4.4 Giao diện truyền thông 9

5. Các yêu cầu phi chức năng khác 10

5.1 Yêu cầu hiệu năng 10

5.2 Yêu cầu an toàn 11

5.3 Yêu cầu bảo mật 11

5.4 Thuộc tính chất lượng phần mềm 12

6. Các yêu cầu khác 13

Phụ lục A: Thuật ngữ 13

Phụ luc B: Mô hình phân tích 13

Phụ lục C: Danh sách vấn đề 14

**1.Giới thiệu**

**1.1 Mục đích**

Hệ thống được thiết kế để theo dõi,dự báo mưa. Ứng dụng sẽ hiển thị thông số thời tiết, hiển thị cho người dùng các thông số, giúp duy trì các điều kiện môi trường phù hợp.

**1.2 Quy ước**

Mục này định nghĩa các thuật ngữ và quy ước sử dụng trong tài liệu để đảm bảo thống nhất và dễ hiểu.

- ESP32: Vi điều khiển tích hợp Wi-Fi và Bluetooth.

- IoT: Internet of Things, kết nối và trao đổi dữ liệu qua mạng Internet.

- Blynk: Nền tảng IoT hỗ trợ giao diện di động.

- Nhiệt độ: Được đo bằng độ C (°C).

- Độ ẩm: Được đo bằng %RH (Relative Humidity - độ ẩm tương đối).

**1.3 Đối tượng và hướng dẫn đọc**

Hệ thống này được thiết kế để phục vụ nhiều nhóm người dùng khác nhau.

\* Đối tượng chính:

- Người dùng cá nhân: Chủ nhà, nhân viên văn phòng, người cần kiểm soát môi trường sống.

- Quản lý cơ quan : Giám sát nhiệt độ, độ ẩm và đọ ẩm đất trong khu vực làm việc.

\* Gợi ý đọc tài liệu:

Lập trình viên: Tập trung vào các yêu cầu kỹ thuật và API của hệ thống.

Nhà phát triển IoT: Chú ý đến giao thức kết nối cảm biến và dữ liệu thời gian thực.

Kỹ thuật viên lắp đặt: Tìm hiểu về phần cứng, kết nối thiết bị.

Người dùng phổ thông: Quan tâm đến cách sử dụng và giao diện ứng dụng.

**1.4 Phạm vi dự án**

Mô tả những gì hệ thống sẽ cung cấp và giới hạn của nó.

Phạm vi của hệ thống:

- Thiết bị IoT: Sử dụng cảm biến để đo nhiệt độ độ ẩm và cảm biến đất

- Ứng dụng di động: Hiển thị dữ liệu thời gian thực và gửi thông báo.

- Kết nối không dây: Gửi dữ liệu từ thiết bị đến ứng dụng qua Wi-Fi.

**1.5 Tài liệu tham khảo**

Một số tài liệu tham khảo có thể bao gồm:

- Sách, tài liệu về IoT: Các nguyên lý về cảm biến và giao tiếp không dây.

- Hướng dẫn sử dụng cảm biến nhiệt độ & độ ẩm:

DHT11/DHT22 (cảm biến nhiệt độ & độ ẩm phổ biến).

BME280 (cảm biến chính xác cao, hỗ trợ áp suất khí quyển).

- Tài liệu về lập trình ứng dụng di động (Android/iOS).

- Hướng dẫn bảo mật dữ liệu IoT để đảm bảo an toàn cho người dùng.

**2. Mô tả tổng thể**

**2.1 Quan điểm**

Hệ thống giám sát nhiệt độ và độ ẩm đưa ra dự báo giúp người dùng theo dõi và kiểm soát môi trường trong phòng từ xa thông qua ứng dụng di động.

* Lợi ích chính:

- Giám sát liên tục: Hiển thị dữ liệu theo thời gian thực.

- Cảnh báo nhanh chóng: Khi nhiệt độ hoặc độ ẩm vượt ngưỡng cài đặt, hệ thống sẽ gửi thông báo ngay lập tức.

- Dễ sử dụng: Ứng dụng có giao diện đơn giản, dễ tiếp cận với mọi đối tượng.

* Ví dụ về ứng dụng thực tế:

- Nhà ở: Giúp theo dõi môi trường sống, đặc biệt hữu ích

- Văn phòng, công ty: Giúp duy trì môi trường làm việc thoải mái.

- Bệnh viện, phòng thí nghiệm: Kiểm soát nhiệt độ, độ ẩm để bảo vệ sức khỏe và đảm bảo điều kiện lưu trữ dược phẩm.

- Kho hàng, nhà xưởng: Đảm bảo điều kiện bảo quản hàng hóa như thực phẩm, thiết bị điện tử, thuốc men.

**2.2 Tính năng**

Hệ thống cung cấp các tính năng sau:

- Theo dõi nhiệt độ và độ ẩm theo thời gian thực đưa ra dự báo về lượng mưa khả năng mưa đồng thời cho biết giá trị về nhiệt độ và độ ẩm.Cảm biến đo nhiệt độ và độ ẩm, gửi dữ liệu đến ứng dụng di động và hiển thị dữ liệu dưới dạng số hoặc biểu đồ để dễ theo dõi.

Cài đặt ngưỡng cảnh báo tùy chỉnh

- Người dùng có thể đặt ngưỡng tối thiểu và tối đa cho nhiệt độ, độ ẩm.

- Hệ thống sẽ tự động giám sát và so sánh với giá trị cài đặt.

**2.3 Lớp người dùng**

Xác định các nhóm người dùng chính của hệ thống và đặc điểm của họ.

|  |  |
| --- | --- |
| Lớp người dùng | Đặc điểm |
| Người dùng phổ thông | Sử dụng ứng dụng để theo dõi môi trường sống, không cần hiểu sâu về kỹ thuật. |
| Quản lý cơ quan, văn phòng | Dùng hệ thống để đảm bảo môi trường làm việc tốt nhất. |
| Nhân viên kỹ thuật, bảo trì | Quan tâm đến dữ liệu nhiệt độ, độ ẩm để bảo trì thiết bị. |
| Lập trình viên | Phát triển và mở rộng hệ thống theo yêu cầu của khách hàng. |

**2.4 Môi trường hoạt động**

Hệ thống hoạt động dựa trên phần cứng, phần mềm và kết nối mạng.

* Phần cứng:

- Cảm biến nhiệt độ & độ ẩm (DHT22, BME280…).

- Vi điều khiển (ESP8266, ESP32, Raspberry Pi…).

- Bộ nguồn đảm bảo hoạt động liên tục.

* Phần mềm:

- Ứng dụng di động Android/iOS để hiển thị dữ liệu.

- Nền tảng đám mây để lưu trữ và xử lý dữ liệu.

* Kết nối mạng:

-Hệ thống sử dụng Wi-Fi để truyền dữ liệu từ thiết bị đến ứng dụng.

-Yêu cầu mạng Internet ổn định để đảm bảo thông báo nhanh chóng.

**2.5 Hạn chế về thiết kế và triển khai**

Mô tả các yếu tố có thể ảnh hưởng đến hiệu suất hệ thống.

* Độ chính xác của cảm biến:

- Cảm biến nhiệt độ và độ ẩm có sai số nhất định (±0.5°C, ±2-5%RH).

- Sai số có thể bị ảnh hưởng bởi môi trường như độ ẩm cao, nhiệt độ thay đổi đột ngột.

* Tính ổn định của chương trình:

- khi đưa ra dự báo cần xác định đúng giá trị vào để đưa ra phần mêm tính toán và chính xác nhất đồng thời tính đồng bộ là ổn định cũng cần nâng cao.

* Tốc độ kết nối Internet:

- Nếu mạng yếu, dữ liệu có thể bị trễ hoặc không hiển thị kịp thời.

- Ứng dụng cần cơ chế lưu trữ dữ liệu tạm thời khi mất kết nối.

**2.6 Tài liệu hướng dẫn sử dụng**

Người dùng sẽ được cung cấp tài liệu hướng dẫn chi tiết để cài đặt và sử dụng hệ thống.

* Tài liệu hướng dẫn bao gồm:

- Cách cài đặt và kết nối thiết bị.

- Cách sử dụng ứng dụng di động.

- Cách đặt ngưỡng cảnh báo.

- Cách kiểm tra và bảo trì thiết bị.

**2.7 Giả định và sự phụ thuộc**

\* Liệt kê các yếu tố bên ngoài có thể ảnh hưởng đến hệ thống.

- Người dùng phải cài ứng dụng để nhận thông báo.

- Thiết bị đo cần được cấp nguồn liên tục

- Nếu mất điện hoặc hết pin, thiết bị sẽ không hoạt động.

- Dữ liệu cần được truyền qua Wi-Fi để hiển thị trên ứng dụng.

**3. Tính năng hệ thống**

**3.1 Thu thập và lưu trữ dữ liệu**

Mô tả chức năng:

Hệ thống sẽ thu thập dữ liệu về nhiệt độ và độ ẩm từ cảm biến, sau đó lưu trữ vào cơ sở dữ liệu để hiển thị và phân tích.

Quy trình hoạt động:

1. Cảm biến đo nhiệt độ độ ẩm và độ ẩm đất theo thời gian thực.
2. Vi điều khiển (ESP32, Raspberry Pi, v.v.) nhận dữ liệu từ cảm biến.
3. Dữ liệu được gửi đến máy chủ đám mây thông qua Wi-Fi.
4. Lưu trữ dữ liệu vào cơ sở dữ liệu để truy xuất và phân tích sau này.

Yêu cầu hệ thống:

* Cảm biến cần có độ chính xác cao.
* Dữ liệu phải được cập nhật theo chu kỳ (ví dụ: 5 giây/lần hoặc 1 phút/lần).
* Kết nối Internet phải ổn định để đảm bảo dữ liệu không bị mất.
* Hệ thống cần có cơ chế lưu trữ tạm thời khi mất mạng.

**3.2 Nội dung hệ thống hiển thị**

Màn hình: Sử dụng màn hình IPS TFT LCD 1.47 inch ST7789 với độ phân giải 172x320 pixel, cho màu sắc rõ nét và góc nhìn rộng.

**3.3 Lịch sử và phân tích dữ liệu (Tính năng mở rộng)**

- Mô tả chức năng:

Hệ thống cho phép người dùng xem lại lịch sử nhiệt độ và độ ẩm theo từng khoảng thời gian.

- Quy trình hoạt động:

1. Người dùng chọn khoảng thời gian muốn xem (1 ngày, 1 tuần, 1 tháng).
2. Ứng dụng truy vấn dữ liệu từ máy chủ và hiển thị dưới dạng biểu đồ.
3. Người dùng có thể xuất dữ liệu dưới dạng file Excel/PDF để phân tích thêm.

Yêu cầu hệ thống:

* Cơ sở dữ liệu phải lưu trữ lịch sử ít nhất 3 tháng - 1 năm.
* Giao diện hiển thị biểu đồ trực quan, dễ hiểu.
* Cho phép tải xuống báo cáo dữ liệu.

**3.4 Điều khiển thiết bị từ xa (Tính năng mở rộng)**

- Mô tả chức năng:

Người dùng có thể điều khiển thiết bị từ xa dựa trên dữ liệu nhiệt độ và độ ẩm.

Ví dụ về ứng dụng:

* Bật/tắt máy lạnh khi nhiệt độ vượt quá ngưỡng.
* Bật quạt hút ẩm nếu độ ẩm quá cao.
* Kết hợp với hệ thống nhà thông minh để tự động điều chỉnh điều kiện môi trường.

- Yêu cầu hệ thống:

* Hệ thống phải hỗ trợ công nghệ IoT để điều khiển thiết bị.
* Đảm bảo độ trễ thấp để phản hồi nhanh.
* Giao diện phải đơn giản, dễ thao tác.

**4 - Yêu cầu về giao diện bên ngoài**

**4.1 Giao diện người dùng**

Mô tả:

Giao diện ứng dụng di động cần được thiết kế trực quan, dễ sử dụng để đảm bảo ngay cả người không có kiến thức kỹ thuật cũng có thể thao tác dễ dàng.

Yêu cầu giao diện:

* Thiết kế đơn giản: Giao diện sạch sẽ, dễ đọc, sử dụng màu sắc phù hợp để phân biệt các mức nhiệt độ/độ ẩm.
* Hiển thị thông tin thời gian thực:

- Nhiệt độ và độ ẩm được cập nhật theo thời gian thực.

- Biểu đồ lịch sử dữ liệu (tuần/tháng).

* Hiển thị rõ ràng:

- hệ thống hiển thị trên giao diện chính.

**4.2 Giao diện phần cứng**

Mô tả:

Phần cứng bao gồm thiết bị đo nhiệt độ, độ ẩm (cảm biến), vi điều khiển và module Wi-Fi để truyền dữ liệu đến ứng dụng.

Yêu cầu thiết bị phần cứng:

* Thiết kế nhỏ gọn, dễ lắp đặt.
* Cảm biến nhiệt độ và độ ẩm có độ chính xác cao (±0.5°C đối với nhiệt độ, ±3%RH đối với độ ẩm).
* Có màn hình nhỏ (tùy chọn): Hiển thị nhiệt độ, độ ẩm ngay trên thiết bị.
* Kết nối Wi-Fi ổn định: Hỗ trợ Wi-Fi 2.4GHz để gửi dữ liệu nhanh chóng.
* Nguồn điện:

- Có thể dùng pin sạc hoặc nguồn điện trực tiếp.

- Nếu dùng pin, cần có cảnh báo pin yếu.

* Khả năng chống chịu môi trường:

-Nếu sử dụng ngoài trời, thiết bị cần có khả năng chống nước, chống bụi (chuẩn IP65 trở lên).

-Nếu đặt trong nhà kho hoặc phòng lạnh, cần chịu được nhiệt độ thấp.

**4.3 Giao diện phần mềm**

Mô tả:

Giao diện phần mềm là các API hoặc giao thức truyền thông mà ứng dụng di động sẽ sử dụng để nhận dữ liệu từ hệ thống IoT.

Yêu cầu phần mềm:

* Ứng dụng di động (Android/iOS):

-Được lập trình bằng Flutter, React Native hoặc Swift/Kotlin để hỗ trợ đa nền tảng.

- Kết nối với server qua REST API hoặc MQTT để nhận dữ liệu từ thiết bị IoT.

- Lưu trữ dữ liệu cục bộ để xem lại khi mất mạng.

* Máy chủ backend:

- Hỗ trợ Node.js, Python (Flask/Django), hoặc Firebase.

- Xử lý dữ liệu nhiệt độ/độ ẩm và gửi thông báo khi cần thiết.

- Lưu trữ dữ liệu trên cloud database (MongoDB, Firebase, PostgreSQL).

* API giao tiếp:

- Cung cấp REST API hoặc MQTT protocol để ứng dụng nhận dữ liệu từ cảm biến.

- Có endpoint để lấy dữ liệu lịch sử, cập nhật cài đặt ngưỡng cảnh báo.

**4.4 Giao diện truyền thông**

Mô tả:

Giao diện truyền thông xác định cách thiết bị IoT và ứng dụng di động trao đổi dữ liệu.

Yêu cầu giao diện truyền thông:

* Giao thức truyền dữ liệu:

- MQTT: Thích hợp cho IoT, tiết kiệm băng thông.

- HTTP/REST API: Dùng khi cần lưu trữ dữ liệu lịch sử.

* Bảo mật dữ liệu:

- Dữ liệu truyền đi phải mã hóa (SSL/TLS) để bảo vệ thông tin người dùng.

- Đăng nhập qua OAuth2 hoặc JWT token để đảm bảo tính bảo mật.

* Tốc độ truyền tải:

- Dữ liệu phải cập nhật tối đa 2 giây/lần để đảm bảo phản hồi nhanh.

- Khi mất mạng, hệ thống cần lưu dữ liệu tạm thời và gửi lại sau khi kết nối phục hồi.

**5 - Các yêu cầu không chức năng khác**

**5.1 Yêu cầu về hiệu suất**

**Mô tả:**

Ứng dụng và hệ thống IoT phải hoạt động nhanh chóng, ổn định, đảm bảo trải nghiệm tốt cho người dùng.

Yêu cầu cụ thể:

* Tốc độ cập nhật dữ liệu:

- Cảm biến gửi dữ liệu lên server mỗi 2-5 giây.

- Ứng dụng di động cập nhật dữ liệu liên tục khi mở.

* Khả năng xử lý dữ liệu:

- Server phải xử lý được ít nhất 1000 thiết bị kết nối đồng thời.

- Cơ sở dữ liệu phải lưu trữ được dữ liệu lịch sử ít nhất 6 tháng.

* Tối ưu pin (nếu dùng pin):

- Thiết bị cảm biến cần chế độ tiết kiệm năng lượng, chỉ gửi dữ liệu khi có thay đổi lớn.

**5.2 Yêu cầu về an toàn**

Mô tả:

Hệ thống cần đảm bảo dữ liệu chính xác và không gây rủi ro cho người dùng.

Yêu cầu cụ thể:

* Bảo vệ dữ liệu đo lường:

- Cảm biến phải có hiệu chuẩn định kỳ để đảm bảo đo lường chính xác.

- Dữ liệu không bị thay đổi sai lệch trong quá trình truyền tải.

* Thông báo lỗi:

- Khi cảm biến hỏng hoặc mất kết nối, ứng dụng phải cảnh báo ngay.

* Giới hạn truy cập:

- Chỉ những người dùng được cấp quyền mới có thể xem/cài đặt ngưỡng cảnh báo.

* Ngăn ngừa sai lệch thông tin:

- Ứng dụng kiểm tra dữ liệu bất thường (ví dụ: nhiệt độ tăng đột ngột từ 25°C lên 100°C).

**5.3 Yêu cầu bảo mật**

Mô tả:

Hệ thống phải bảo vệ dữ liệu người dùng và thiết bị khỏi tấn công mạng, đánh cắp thông tin.

Yêu cầu cụ thể:

* Xác thực tài khoản người dùng:

- Người dùng phải đăng nhập bằng email/số điện thoại + mật khẩu.

- Hỗ trợ OTP hoặc xác thực hai lớp (2FA) nếu cần.

* Mã hóa dữ liệu:

- Dữ liệu truyền từ cảm biến đến server phải mã hóa bằng SSL/TLS.

- Thông tin tài khoản người dùng phải lưu trữ dưới dạng mã băm (bcrypt, SHA-256).

* Bảo vệ thiết bị IoT:

- Thiết bị phải có mật khẩu hoặc khóa mã hóa để ngăn chặn truy cập trái phép.

- Không để cổng giao tiếp mở (SSH, Telnet) trên thiết bị IoT.

* Phát hiện tấn công:

- Server phải có cơ chế giới hạn số lần đăng nhập sai để ngăn brute force.

- Log hệ thống phải lưu lại tất cả các lần truy cập đáng ngờ.

**5.4 Thuộc tính chất lượng phần mềm**

**Mô tả:**

Phần mềm cần đảm bảo các tiêu chí chất lượng như ổn định, dễ sử dụng, có thể mở rộng.

Yêu cầu cụ thể:

* Ổn định:

- Ứng dụng không bị crash khi sử dụng trong thời gian dài.

- Hệ thống tiếp tục hoạt động ngay cả khi server có lỗi nhẹ (có cơ chế backup).

* Khả năng mở rộng:

- Hệ thống phải có thể mở rộng để hỗ trợ 10,000+ thiết bị trong tương lai.

- Dễ dàng nâng cấp để hỗ trợ các loại cảm biến mới.

* Dễ bảo trì:

- Mã nguồn phải viết dễ hiểu, có tài liệu đầy đủ.

- Cấu trúc server dễ nâng cấp, không bị lỗi khi cập nhật.

* Khả năng tương thích:

- Hỗ trợ nhiều loại thiết bị IoT khác nhau nếu cần.

**6. Các yêu cầu khác**

Các yêu cầu đặc biệt khác có thể bao gồm yêu cầu về tích hợp với các hệ thống khác hoặc khả năng mở rộng trong tương lai.

**6.1 Phụ lục A: Thuật ngữ**

- ESP32: Vi điều khiển tích hợp Wi-Fi và Bluetooth.

- IoT: Internet of Things.

- Blynk: Nền tảng IoT cho giao diện di động.

**6.2 Phụ lục B: Mô hình phân tích**

Mô hình phân tích hệ thống sẽ được xây dựng sau khi hoàn thành các yêu cầu chức năng

**6.3 Phụ lục C: Danh sách vấn đề**

< Đây là danh sách động các vấn đề về yêu cầu mở vẫn chưa được giải quyết, bao gồm TBD, quyết định đang chờ xử lý, thông tin cần thiết, xung đột đang chờ giải quyết, v.v.>