

Viet Nam national university Ho Chi Minh city
University of Technology
Faculty of Computer Science and Engineering



Multidisciplinary Project (CO3019)

Assignment (Semester: 222, Duration: 12 weeks)

“YOLO:FARM”

Advisor: Mai Đức Trung

Team : DADN

Student(s): Lê Huỳnh Đăng Khoa - 2011420
Tô Duy Tân - 2012017
Nguyễn Văn Thịnh - 2014603
Lý Chánh Tín - 2012211

Ho Chi Minh City , 11th June 2023



Contents

List of Figures	3
List of Tables	4
Member list & Workload	5
DOCUMENT HISTORY	6
1 Requirement elicitation	7
1.1 Tổng quan đề tài	7
1.1.1 Vấn đề đặt ra	7
1.1.2 Bối cảnh dự án	7
1.1.3 Mô tả dự án	7
1.1.4 Xác định yêu cầu	8
1.1.4.a Yêu cầu chức năng	8
1.1.4.b Yêu cầu phi chức năng	9
1.1.5 Nguồn lực sử dụng	9
1.1.5.a Phần cứng	9
1.1.5.b Phần mềm	9
1.1.6 Use-case diagram	9
1.1.6.a Use-case diagram	9
1.1.6.b Mô tả sơ lược các use-case của hệ thống	11
1.2 Use-case scenario	11
1.2.1 Analyze environment status	11
1.2.1.a Display Real-time Environment Data	11
1.2.1.b Detect threshold-exceeded data	12
1.2.2 Manage Irrigation	13
1.2.2.a Send Scheduled Notification	13
1.2.2.b Update Control Sensor Value to Cloud	13
1.2.2.c Create Irrigation Plan	13
1.2.2.d Activate All Plan Control Devices	14
1.2.3 Manage Sensor	15
1.2.3.a Get Data From Server	15
1.2.3.b View Sensor Configuration	15
1.2.3.c Update Sensor Configuration	15
1.2.4 Analyze Data	16
1.2.4.a View Statistics	16
1.2.4.b Calculate Data	17
1.2.4.c Create Data Chart	17
2 System Modeling	18
2.1 Database Design	18
2.2 Develop MVP	18
2.2.1 Manage Sensor & Analyze environment status	18
2.2.2 Manage Irrigation	19
2.2.3 Analyze data	21
2.2.4 Notification	22



3	Architecture Design	23
3.1	Hệ thống nhóm đề xuất	23
3.2	Kiến trúc 3 tầng (Three-layer Architecture)	23
3.3	Mô hình MVC	24
3.4	Lý do nhóm chọn mô hình MVC	25
3.5	Architectural diagram	26
3.6	Kế hoạch dự định hiện thực	27
4	Implementation - Sprint	29
4.1	Giao diện ứng dụng	29
4.1.1	Giao diện phân tích trạng thái của môi trường và quản lý các cảm biến	29
4.1.2	Giao diện quản lý tưới tiêu	31
4.1.3	Giao diện phân tích dữ liệu của hệ thống	33
4.2	Kế hoạch thuyết trình	35



List of Figures

1	Use-case diagram for the whole system	10
2	E-ERD	18
3	Sensor Screen Flow	19
4	Schedule Screen Flow	20
5	Chart Screen Flow	21
6	Notification Screen Flow	22
7	Hệ thống đề xuất	23
8	Three-layer Architecture	24
9	MVC Archietecture pattern	25
10	Architectural diagram	26
11	RESTful API	27
12	MongoDB	28
13	Giao diện Trang chủ phân tích trạng thái môi trường	29
14	Giao diện xem chi tiết và quản lý các cảm biến	30
15	Giao diện lập lịch tưới tiêu	31
16	Giao diện điều chỉnh chế độ tưới nước	32
17	Giao diện các thông số	33
18	Giao diện chi tiết thông số của cây trồng	34
19	Giao diện theo ngày của thông số	35



List of Tables

1	Member list & Workload	5
2	Document history	6
3	Bảng danh sách kèm mô tả sơ lược các use-case trong hệ thống	11
4	Display Real-time Environment Data use-case scenario	11
5	Display Threshold-exceeded Data use-case scenario	12
6	Send Scheduled Notification use-case scenario	13
7	Update Control Sensor Value to Cloud use-case scenario	13
8	Create Irrigation Plan use-case scenario	14
9	Activate All Plan Control Devices use-case scenario	14
10	Use-case scenario of Get data from server	15
11	View Sensor Configuration use-case scenario	15
12	Configure Sensor Setting use-case scenario	16
13	Use-case scenario of View Statistics	16
14	Use-case scenario of Calculate Data	17
15	Use-case scenario of Create Data Chart	17



MEMBER LIST & WORKLOAD

No.	Fullname	Student ID	Problems	Percentage of work
1	Lê Huỳnh Đăng Khoa	2011420	Manage Irrigation	100%
2	Tô Duy Tân	2012017	Analyze Data	100%
3	Nguyễn Văn Thịnh	2014603	Manage Sensor	100%
4	Lý Chánh Tín	2012211	Analyze environment status	100%

Bảng 1: Member list & Workload



DOCUMENT HISTORY

Date	Version	Changes	Person in charge
17 Feb	1.0	Hoàn thành cơ bản phần mô tả dự án	All
22 Feb	1.0.1	Hoàn thành phần Use-Case Diagram	All
24 Feb	1.0.2	Hoàn thành Scenario cho Use-Case Diagram	All
3 Mar	1.0.3	Hộp và phân chia công việc phần design UI	All

Bảng 2: Document history

1 Requirement elicitation

1.1 Tổng quan đề tài

1.1.1 Vấn đề đặt ra

Tại Việt Nam, nông nghiệp là một ngành kinh tế chủ lực và tạo lượng lớn việc làm cho lao động ở khu vực nông thôn. Thế nhưng ngành này đang phải đối mặt trực diện và chịu nhiều thiệt hại từ những tác động tiêu cực của biến đổi khí hậu như nước biển dâng, lũ lụt, hạn hán, xâm nhập mặn, thời tiết xấu, làm ảnh hưởng nghiêm trọng đến sản xuất nông nghiệp, vốn dựa nhiều vào các hoạt động thủ công.

Sự phát triển của nông nghiệp công nghệ (hay còn gọi là nông nghiệp 4.0) đã giúp giải quyết đáng kể bài toán chi phí, lao động, hiệu quả trong sản xuất và đáp ứng tốt hơn nhu cầu thực phẩm trong nước. Trong phạm vi dự án này, nhóm đề xuất, thiết kế và hiện thực một hệ thống IoT để quản lý trang trại cây ăn quả nhằm giải quyết những vấn đề cơ bản của mô hình nông nghiệp truyền thống như đã đề cập.

1.1.2 Bối cảnh dự án

- Dự án xây dựng ứng dụng quản lý trang trại thông minh trồng cây ăn quả ở Đồng bằng sông Cửu Long diễn ra trong một thời điểm mà ngành nông nghiệp đang chịu nhiều áp lực từ sự thay đổi khí hậu nhất là tình trạng xâm nhập mặn gây thiếu nước ngọt cho việc tưới tiêu và khó khăn trong sản xuất. Đặc biệt, trong vùng Đồng bằng sông Cửu Long, với diện tích trồng cây ăn quả lớn, các nhà nông gặp nhiều khó khăn trong việc quản lý và theo dõi tình trạng cây trồng.
- Các nhà nông truyền thống phải thường xuyên kiểm tra tình trạng của cây trồng, tưới nước, chăm sóc và phòng chống sâu bệnh, tuy nhiên, việc này mất nhiều thời gian và công sức. Bên cạnh đó, nước tưới thường được dùng quá nhiều (gây lãng phí nước) hoặc quá ít, khiến cho năng suất cây trồng giảm sút hoặc gặp sự cố do tác động của thời tiết, sâu bệnh và mất nước.
- Với sự ra đời của ứng dụng quản lý trang trại thông minh, việc quản lý và theo dõi tình trạng cây trồng trở nên dễ dàng hơn. Ứng dụng này sử dụng các thiết bị IoT để thu thập dữ liệu và đưa ra các lời khuyên dựa trên dữ liệu thu thập được, giúp người nông dân giảm thiểu thời gian và công sức quản lý trang trại, đồng thời tăng năng suất và chất lượng sản phẩm. Ngoài ra, ứng dụng còn giúp người nông dân tiết kiệm nước tưới, chủ động trong việc tưới tiêu giúp nông dân quản lý và sử dụng tài nguyên nước một cách hiệu quả hơn, giảm thiểu thiệt hại cho nông nghiệp do xâm nhập mặn và thay đổi khí hậu gây ra, giảm thiểu tác động của việc sử dụng hóa chất trong quá trình trồng trọt, góp phần bảo vệ môi trường và phát triển bền vững ngành nông nghiệp.

Vì vậy nhóm đã chọn thực hiện ứng dụng quản lý trang trại thông minh trên đối tượng là vườn trồng cây ăn quả (Sầu Riêng) ở Đồng bằng Sông Cửu Long là bối cảnh cho dự án này.

1.1.3 Mô tả dự án

Mục tiêu của dự án này là đề xuất một hệ thống quản lý nông trại cây ăn quả thông minh bằng một hệ thống IoT:

- Hiện nay, với sự phát triển của công nghệ IoT, việc ứng dụng IoT vào nông nghiệp đang được nhiều nhà nông quan tâm và áp dụng. Việc xây dựng một hệ thống IoT cho một mô hình nông trại có thể giúp tăng năng suất, giảm bớt thời gian và công sức quản lý. Hệ thống này có khả năng tự động hoá các quy trình nông nghiệp, giúp tiết kiệm chi phí, tăng năng suất và giảm thiểu sự phụ thuộc vào con người. Một hệ thống IoT nông trại thông minh có thể được thiết kế để giám sát và điều khiển hoạt động của nông trại, từ việc theo dõi độ ẩm, nhiệt độ, độ sáng, tưới cây, cho đến việc quản lý sản phẩm thu hoạch. Dữ liệu được thu thập từ các cảm biến trên nông trại sẽ được gửi đến một trung tâm quản lý thông minh, nơi mà hệ thống sẽ xử lý và phân tích dữ liệu để đưa ra các thông tin hữu ích cho người quản lý nông trại.
- Sử dụng các phương tiện kết nối như máy tính, điện thoại, máy tính bảng, người dùng có thể theo dõi trực tiếp và quản lý nông trại của mình ở bất kỳ đâu, bất kỳ lúc nào. Các hệ thống IoT cho nông trại cũng có thể được kết hợp với hệ thống trí tuệ nhân tạo để phân tích dữ liệu và đưa ra các dự đoán về tình trạng nông trại, từ đó người quản lý có thể đưa ra các quyết định đúng đắn hơn và tối ưu hóa hoạt động của nông trại.

1.1.4 Xác định yêu cầu

1.1.4.a Yêu cầu chức năng

- Giám sát các điều kiện môi trường
 - Đưa ra cảnh báo khi dữ liệu vượt ngưỡng cho phép: khi nhiệt độ quá cao hoặc khi độ ẩm trong đất quá thấp, kiểm soát các thông số môi trường
- Quản lý các thiết bị cảm biến
 - Lưu trữ dữ liệu gửi về từ các thiết bị
 - Thu thập dữ liệu về các điều kiện môi trường từ các thiết bị cảm biến
 - * Nhiệt độ
 - * Độ ẩm không khí
 - * Ánh sáng
- Kiểm soát tưới nước thông minh
 - Đặt hẹn tưới tiêu theo thời gian và ngày mà người quản lý định sẵn
 - Cho phép người quản lý kích hoạt hệ thống tưới bằng các thiết bị thông minh
 - Gửi thông báo cần tưới nước cho người quản lý từ dữ liệu thu thập được từ module giám sát môi trường
- Phân tích dữ liệu
 - Tính toán thời gian tăng trưởng của cây trồng dựa vào cảm biến ánh sáng.
 - Biểu diễn dưới dạng đồ thị trực quan từ các dữ liệu quan trắc

1.1.4.b Yêu cầu phi chức năng

- Tính sẵn sàng: khi hệ thống gặp sự cố, kết nối lại trong tối đa 1 phút. Thời gian gặp sự cố không quá 15 phút mỗi ngày.
- Tính thân thiện: người dùng có thể dễ dàng thao tác hầu hết các chức năng trong vòng 15 phút.
- Tính tương thích: Giao diện quản lý và giám sát có thể cài đặt dễ dàng trên IOS và Android
- Hiệu năng: Cập nhật dữ liệu nhanh chóng dưới 5s

1.1.5 Nguồn lực sử dụng

1.1.5.a Phần cứng

- Cảm biến nhiệt độ
- Màn hình LCD 16x2
- Cảm biến độ ẩm đất
- Cảm biến ánh sáng
- Yolo:Bit
- Đèn 4 Led RGB
- Động cơ bơm nước
- Động cơ Servo
- Ống dẫn nước
- Công tắc Relay

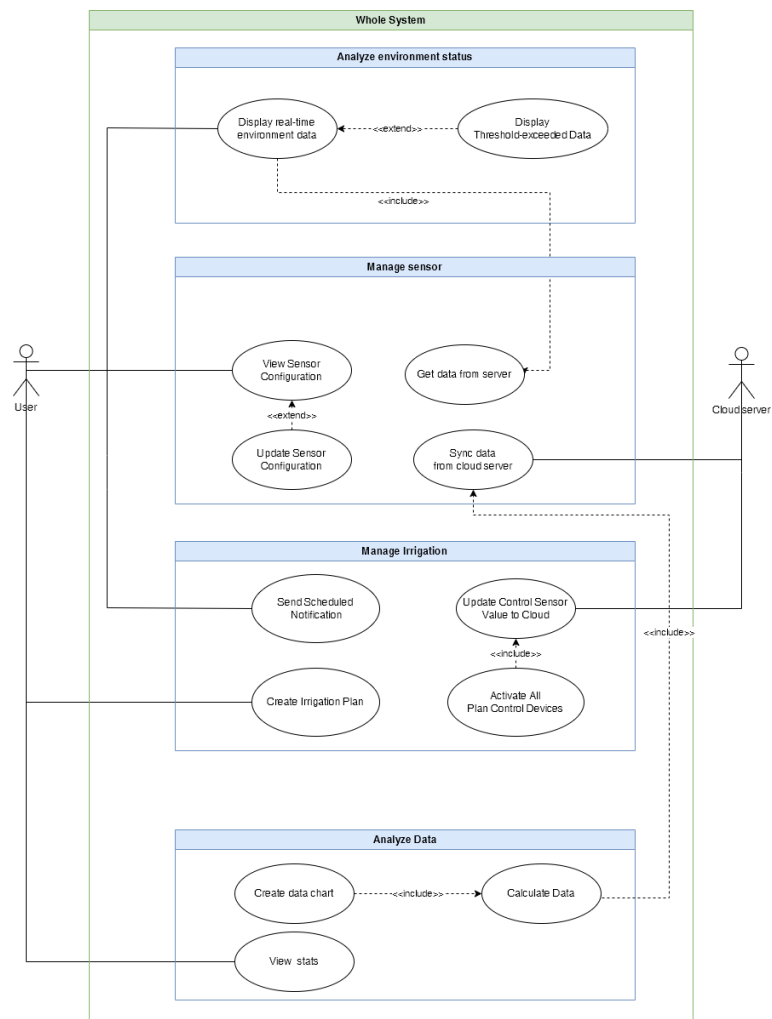
1.1.5.b Phần mềm

- Framework: React Native
- Database: MongoDB

1.1.6 Use-case diagram

1.1.6.a Use-case diagram

Xem chi tiết use-case diagram tại [đây](#)



Hình 1: Use-case diagram for the whole system

1.1.6.b Mô tả sơ lược các use-case của hệ thống

No.	Use case	Job description
1	Display real-time environment data	Dùng để xem các thông báo quan trắc môi trường theo thời gian thực
2	Display Threshold-exceeded Data	Hiển thị thông số cảnh báo khi giá trị quan trắc vượt ngưỡng
3	View Sensor Configuration	Xem chi tiết cài đặt của các cảm biến
4	Update Sensor Configuration	Cập nhật thông số trên thiết bị cảm biến
5	Get data from server	Lấy thông tin dữ liệu quan trắc từ máy chủ
6	Sync data from cloud server	Đồng bộ dữ liệu từ thiết bị quan trắc vào cơ sở dữ liệu
7	Send Scheduled Notification	Gửi thông báo về lịch tưới nước đã được hẹn giờ
8	Update Control Sensor Value to Cloud	Cập nhật giá trị thiết bị điều khiển lên máy chủ trên Cloud
9	Create Irrigation Plan	Tạo một lịch tưới nước mới
10	Activate All Plan Control Devices	Kích hoạt tất cả các thiết bị trong một lịch tưới theo thời gian định sẵn
11	View stats	Xem dữ liệu quan trắc dưới dạng các biểu đồ
12	Calculate Data	Tính toán dữ liệu cần thiết cho việc tạo biểu đồ
13	Create data chart	Tạo biểu đồ dữ liệu từ các dữ liệu gửi về của cảm biến

Bảng 3: Bảng danh sách kèm mô tả sơ lược các use-case trong hệ thống

1.2 Use-case scenario

1.2.1 Analyze environment status

1.2.1.a Display Real-time Environment Data

Use-Case Name	Display Real-time Environment Data
Actor	User
Trigger	Người dùng nhấn vào biểu tượng hiển thị thông số môi trường trên thanh điều hướng hoặc nhấn nút Đồng bộ tại trang hiển thị thông số môi trường
Description	Cho phép người dùng xem các thông báo về môi trường và giám sát tình trạng cây trồng để chăm sóc kịp thời
Preconditions	Dữ liệu được trả về từ máy chủ của hệ thống thành công
Postconditions	Hệ thống hiển thị chi tiết thông số thiết bị cho người dùng
Normal flow	1. Hệ thống hiển thị dữ liệu theo định dạng đã quy định ra thiết bị của người dùng. 2. Người dùng xem thông các thông số môi trường mới nhất trên màn hình thiết bị *Extension point: Display Threshold-exceeded Data
Alternative flows	None
Exceptions	None

Bảng 4: Display Real-time Environment Data use-case scenario



1.2.1.b Detect threshold-exceeded data

Use-Case Name	Display Threshold-exceeded Data
Actor	User
Trigger	Người dùng tải lại dữ liệu bằng nút đồng bộ trên thiết bị hoặc tải dữ liệu lần đầu tiên
Description	Hiển thị các giá trị môi trường vượt quá ngưỡng đã cài đặt với định dạng khác biệt, dễ nhận diện
Preconditions	None
Postconditions	Dữ liệu vượt quá ngưỡng được hiển thị tới người dùng với định dạng dễ dàng nhận biết
Normal flow	<ol style="list-style-type: none">1. Hệ thống kiểm tra dữ liệu được đã được lưu trữ tại cơ sở dữ liệu.2. Hệ thống so sánh dữ liệu nhận được với ngưỡng đã thiết đặt3. Hệ thống hiển thị các thông số vượt ngưỡng bằng màu đỏ chữ in đậm lên màn hình
Alternative flows	None
Exceptions	None

Bảng 5: Display Threshold-exceeded Data use-case scenario

1.2.2 Manage Irrigation

1.2.2.a Send Scheduled Notification

Use-Case Name	Send Scheduled Notification
Actor	User
Trigger	Hệ thống tự động kiểm tra và tạo thông báo khi gần đến lịch tưới nước đã hẹn.
Description	Hệ thống gửi thông báo trước thời gian tưới nước tự động cho cây trồng
Preconditions	Các thiết bị tưới nước đã được thiết lập và người dùng kết nối thành công vào ứng dụng
Postconditions	Lịch tưới nước được gửi cho người dùng trước khoảng thời gian mà người dùng đã thiết lập.
Normal flow	1. Hệ thống lấy thông tin lịch các lịch tưới nước đã tạo từ cơ sở dữ liệu 2. Hệ thống kiểm tra và xác định các lịch tưới sắp diễn ra 3. Hệ thống tạo thông báo về các lịch tưới mới và lưu trữ tại cơ sở dữ liệu
Alternative flows	None
Exceptions	None

Bảng 6: Send Scheduled Notification use-case scenario

1.2.2.b Update Control Sensor Value to Cloud

Use-Case Name	Update Control Sensor Value to Cloud
Actor	User
Trigger	Người dùng nhấn vào giao diện đặt lịch trên thanh điều hướng
Description	Tính năng này cho phép người dùng điều khiển thiết bị tưới trực tiếp thông qua ứng dụng
Preconditions	Các thiết bị tưới nước đã được thiết lập và người dùng kết nối thành công vào ứng dụng
Postconditions	Máy bơm nước nhận được tín hiệu điều khiển và tưới nước thành công.
Normal flow	1. Hệ thống hiển thị giao diện và danh sách các thiết bị tưới có thể điều khiển 2. Người dùng chọn thiết bị muốn kích hoạt . 3. Hệ thống hiển thị thông tin thiết bị với các thông số đã cài đặt 4. Người dùng nhấn nút tưới nước để gửi tín hiệu 5. Hệ thống gửi tín hiệu đến thiết bị bơm để kích hoạt máy bơm. 6. Ứng dụng hiển thị thông báo tưới nước thành công cho người dùng.
Alternative flows	None
Exceptions	None

Bảng 7: Update Control Sensor Value to Cloud use-case scenario

1.2.2.c Create Irrigation Plan

Use-Case Name	Create Irrigation Plan
Actor	User
Trigger	Người dùng nhấn vào nút thêm mới(Dấu cộng) tại giao diện lên lịch tưới nước
Description	Tính năng này cho phép người dùng lên lịch tưới nước tự động
Preconditions	Các thiết bị tưới nước đã được thiết lập và người dùng kết nối thành công vào ứng dụng
Postconditions	Lịch tưới nước được lưu trữ vào cơ sở dữ liệu và sẽ được dùng cho việc tưới nước tự động theo lịch.
Normal flow	1. Hệ thống hiển thị giao diện đặt lịch theo khu vực và danh sách các thiết bị tưới có thể điều khiển 2. Người dùng chọn thời gian muốn tạo. 3. Người dùng kiểm tra và nhấn nút xác nhận lịch 4. Hệ thống tạo lịch tưới nước mới và lưu trữ vào cơ sở dữ liệu 5. Hệ thống gửi thông báo tạo lịch thành công đến người dùng
Alternative flows	Tại sau bước 2, người dùng muốn hoàn tác việc đặt lịch 1. Người dùng ấn nút quay lại trên màn hình tạo lịch mới 2. Hệ thống quay lại giao diện xem lịch mà không lưu trữ bất kì thông tin nào
Exceptions	None

Bảng 8: Create Irrigation Plan use-case scenario

1.2.2.d Activate All Plan Control Devices

Use-Case Name	Activate All Plan Control Devices
Actor	
Trigger	Hệ thống tự động kích hoạt khi đến thời gian tưới trên lịch đã tạo
Description	Tính năng này cho phép tự động tưới nước theo lịch đã được hẹn bởi người dùng
Preconditions	Các thiết bị tưới nước đã được đăng kí vào hệ thống
Postconditions	Các máy bơm được điều khiển và thực hiện tưới nước tự động theo lịch đã định
Normal flow	1. Hệ thống lấy thông tin các lịch tưới nước đã lưu trữ 2. Hệ thống kiểm tra các lịch tưới đã đến thời gian. 3. Hệ thống thực hiện use-case "Update Control Sensor Value to Cloud" để đặt trạng thái tưới cho các cảm biến tương ứng 4. Hệ thống thực hiện lại use-case "Update Control Sensor Value to Cloud" một lần nữa khi hết thời gian để đặt lại trạng thái dừng cho các cảm biến điều khiển
Alternative flows	None
Exceptions	Tại bước 3, cảm biến không nhận được tín hiệu 1. Hệ thống gửi lại tín hiệu điều khiển sau khi hết thời gian chờ . 2. Quay lại bước 2 của normal flow.

Bảng 9: Activate All Plan Control Devices use-case scenario

1.2.3 Manage Sensor

1.2.3.a Get Data From Server

Use-Case Name	Get Data From Server
Actor	User
Trigger	Người dùng nhấn vào biểu tượng trang chủ trên thanh điều hướng
Description	Hệ thống lấy dữ liệu về thông số môi trường đã được lưu trữ
Preconditions	Truy cập được vào Server để thực hiện lấy dữ liệu
Postconditions	Dữ liệu được gửi đến thiết bị của người dùng để hiển thị
Normal flow	1. Hệ thống gửi yêu cầu lấy dữ liệu xuống máy chủ 2. Máy chủ thực hiện truy vấn từ cơ sở dữ liệu để lấy thông số môi trường mới nhất và gửi đến thiết bị người dùng 3. Dữ liệu được thiết bị của người dùng nhận để tiến hành hiển thị
Alternative flows	None
Exceptions	None

Bảng 10: Use-case scenario of Get data from server

1.2.3.b View Sensor Configuration

Use-Case Name	View Sensor Configuration
Actor	User
Trigger	User nhấn vào nút "View Sensor Information" trên màn hình
Description	Cho phép người dùng xem chi tiết các thông tin về cảm biến
Preconditions	None
Postconditions	Xem được thông tin cụ thể về thuộc tính của từng cảm biến.
Normal flow	1. Hệ thống hiển thị: - Danh sách các cảm biến 2. Người dùng nhấn vào "chi tiết" tương ứng với tên của cảm biến cần xem. 3. Hệ thống hiển thị thông tin từng thuộc tính về cảm biến được chọn.
Alternative flows	None
Exceptions	Tại bước 1, không có cảm biến nào được hiển thị.

Bảng 11: View Sensor Configuration use-case scenario

1.2.3.c Update Sensor Configuration

Use-Case Name	Update Sensor Configuration
Actor	User
Trigger	User nhấn vào nút "Configure Sensor Setting" trên màn hình
Description	Cho phép người dùng cấu hình lại các thuộc tính liên quan đến cảm biến
Preconditions	None
Postconditions	POST-1: Xem được thông tin cụ thể về thuộc tính của từng cảm biến. POST-2: Chính được dữ liệu của từng thuộc tính.
Normal flow	1. Hệ thống hiển thị: - Danh sách các cảm biến 2. Người dùng nhấn vào xem "chi tiết" tương ứng với tên của cảm biến. 3. Hệ thống hiển thị thông tin liên quan đến cảm biến. 4. Người dùng nhấn vào nút "chỉnh sửa". 5. Người dùng nhấn vào nút "xác nhận" để hoàn thành chỉnh sửa.
Alternative flows	None
Exceptions	Tại bước 1, không có cảm biến nào được hiển thị.

Bảng 12: Configure Sensor Setting use-case scenario

1.2.4 Analyze Data

1.2.4.a View Statistics

Use-Case Name	View Statistics
Actor	User
Trigger	User nhấn vào biểu tượng thông số trên màn hình
Description	Cho phép User xem các thông số, biểu đồ của môi trường (độ ẩm, nhiệt độ, ánh sáng) và các lịch sử hoạt động của User
Preconditions	User đã kết nối thành công vào hệ thống
Postconditions	POST-1: User xem được các thông số, biểu đồ của môi trường. POST-2: User xem được lịch sử hoạt động của mình.
Normal flow	1. User nhấn vào biểu tượng thông số trên màn hình 2. Hệ thống hiển thị các thông số, biểu đồ của môi trường (độ ẩm, nhiệt độ, ánh sáng).
Alternative flows	Alternative flow 1: 2.a. User nhấn vào Logs để xem các lịch sử hoạt động. 2.b. Hệ thống hiển thị lịch sử hoạt động của User.
Exceptions	None

Bảng 13: Use-case scenario of View Statistics



1.2.4.b Calculate Data

Use-Case Name	Calculate Data
Actor	None
Trigger	None
Description	Hệ thống tự động tính toán các dữ liệu được nhận từ cảm biến thông qua Cloud Server
Preconditions	Hệ thống kết nối thành công với Cloud Server
Postconditions	POST-1: Các thông số được tính toán bằng các dữ liệu từ cảm biến.
Normal flow	1. Hệ thống nhận dữ liệu từ cảm biến qua Cloud Server 2. Hệ thống tính toán các thông số từ dữ liệu được nhận: <ul style="list-style-type: none">- Độ ẩm trung bình- Nhiệt độ trung bình.- Ánh sáng trung bình.- Số lần tưới nước trung bình.
Alternative flows	None
Exceptions	None

Bảng 14: Use-case scenario of Calculate Data

1.2.4.c Create Data Chart

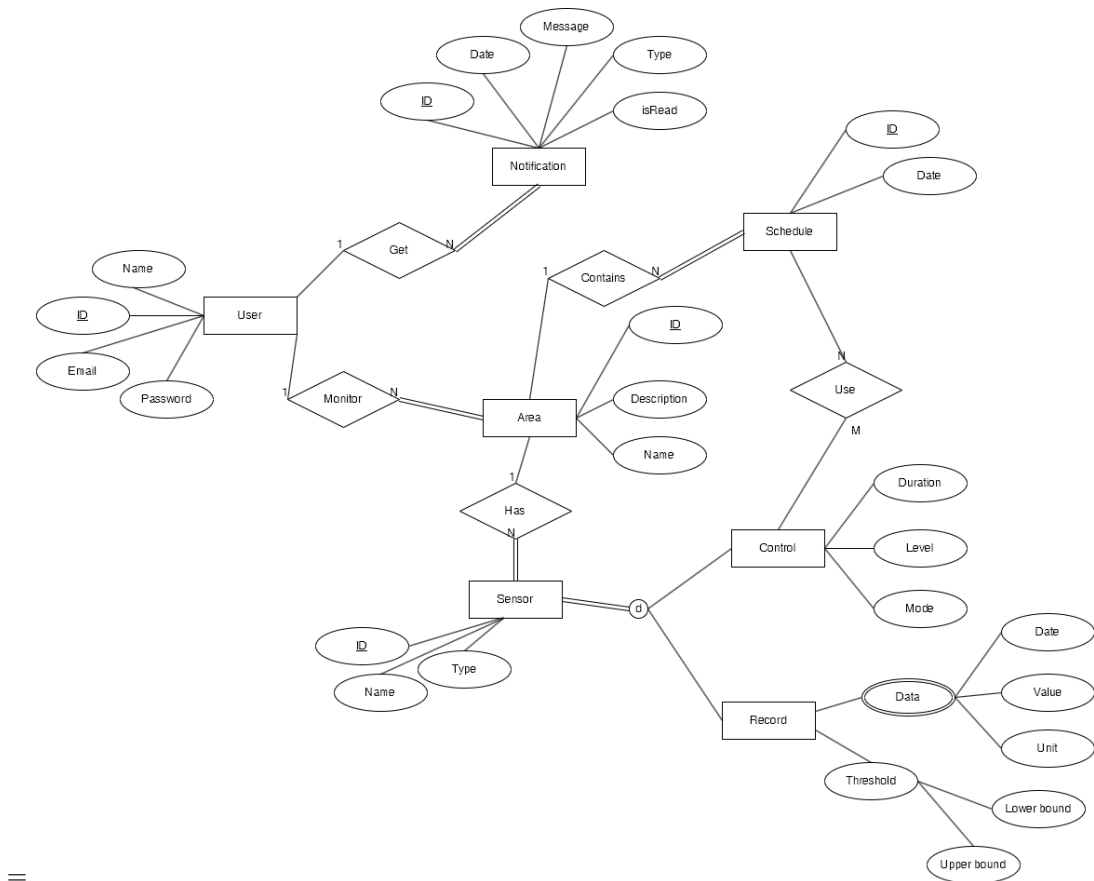
Use-Case Name	Create Data Chart
Actor	None
Trigger	None
Description	Hệ thống tự động vẽ biểu đồ về môi trường từ dữ liệu nhận được từ cảm biến
Preconditions	Hệ thống kết nối thành công với Cloud Server
Postconditions	POST-1: Các biểu đồ được vẽ bằng các dữ liệu từ cảm biến.
Normal flow	1. Hệ thống nhận dữ liệu từ cảm biến qua Cloud Server 2. Hệ thống vẽ biểu đồ từ các thông số từ dữ liệu được nhận: <ul style="list-style-type: none">- Biểu đồ độ ẩm.- Biểu đồ nhiệt độ.- Biểu đồ ánh sáng.
Alternative flows	None
Exceptions	None

Bảng 15: Use-case scenario of Create Data Chart

2 System Modeling

2.1 Database Design

Lược đồ E-ERD mô tả các dữ liệu cần lưu trữ của nhóm:



Hình 2: E-ERD

Xem chi tiết lược đồ tại [đây](#)

2.2 Develop MVP

Nhóm sử dụng [figma](#) làm framework để vẽ UI.

Workspace của nhóm: [workspace link](#)

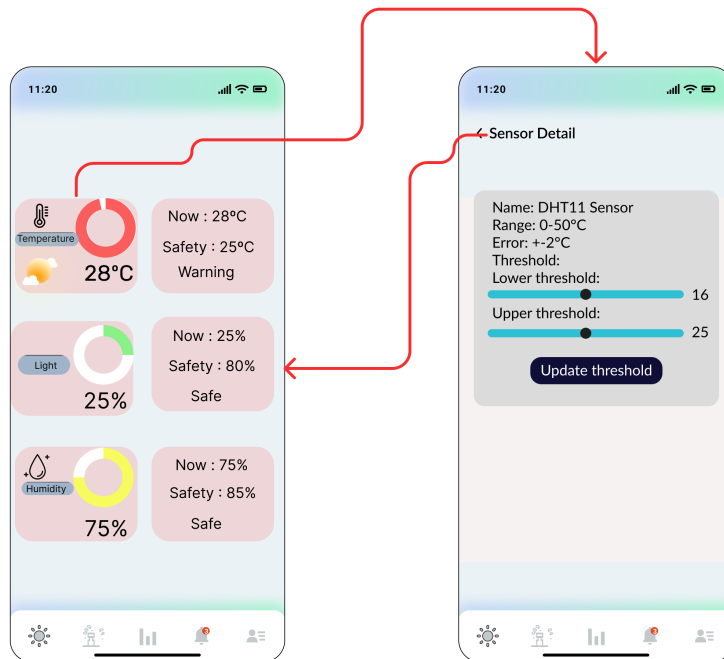
2.2.1 Manage Sensor & Analyze environment status

Trang chủ của hệ thống được mở bằng cách nhấn vào biểu tượng đầu tiên bên trái trên thanh điều hướng. Trang này giúp người dùng xem thông tin các giá trị quan trắc môi trường và đưa ra cảnh báo khi giá trị đo vượt quá ngưỡng cho phép.

Ngoài ra, người dùng có thể điều hướng từ trang này đến trang chi tiết cảm biến để thiết lập

ngưỡng giá trị mới.

Screen Flow cho nhóm tính năng quan trắc như sau:



=

Hình 3: Sensor Screen Flow

Mô tả:

- Tại trang chủ: người dùng nhấn vào chi tiết một thiết bị để đi đến trang cài đặt thông số.
- Tại trang cài đặt: Người dùng nhấn vào nút quay về ở đầu trang để về lại trang chủ. Ngoài ra, người dùng có thể nhấn nút Update threshold để lưu lại thông số thiết bị (Use-case update sensor configuration)

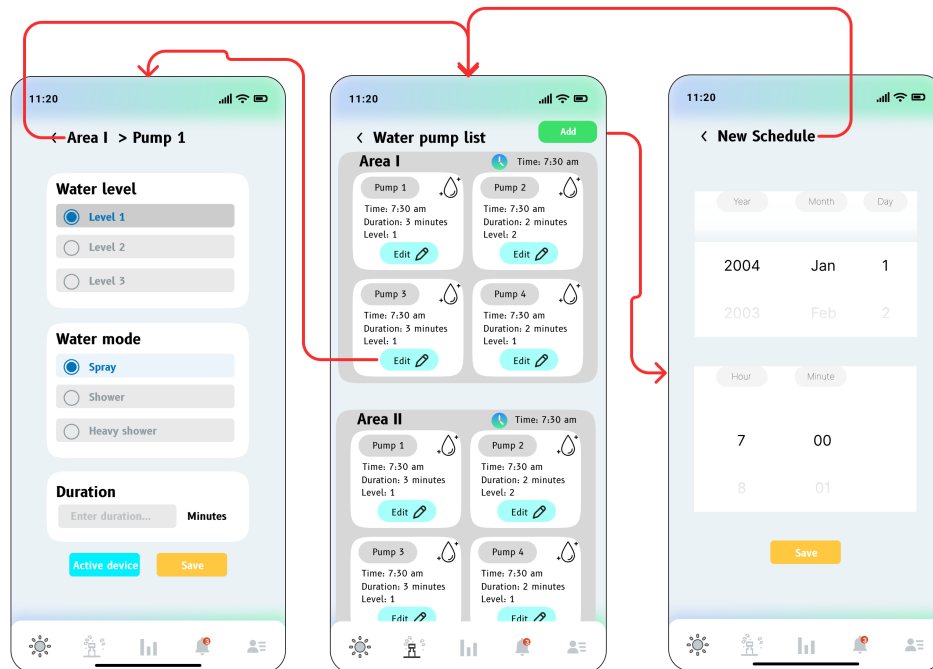
2.2.2 Manage Irrigation

Để thực hiện nhóm các chức năng liên quan đến tưới tiêu, người dùng ấn vào biểu tượng thứ hai tính từ bên trái trên thanh điều hướng.

Nhóm chức năng này gồm có ba màn hình chính:

- Màn hình chứa danh sách các thiết bị đã được lên lịch ứng với từng khu vực
- Màn hình cài đặt chi tiết cho thiết bị
- Màn hình tạo một lịch tưới mới

Screen Flow cho nhóm chức năng quản lý tưới tiêu:



Hình 4: Schedule Screen Flow

Các đối tượng quan trọng:

- Nút Add tại trang danh sách thiết bị: Giúp người dùng chuyển đến trang tạo lịch tưới mới.
- Nút Save tại trang tạo lịch tưới mới: Giúp người dùng xác định thời điểm đã chọn và tạo lịch tưới nước mới.
- Nút Active Schedule tại trang cài đặt thiết bị: Giúp người dùng trực tiếp kích hoạt thiết bị với thông số hiện có
- Nút Save tại trang cài đặt thiết bị: Giúp người dùng lưu thông số đã điều chỉnh trên thiết bị.
- Các nút quay lại ở đầu trang tạo lịch tưới và cài đặt thiết bị: Giúp người dùng quay về trang danh sách các lịch tưới nước.

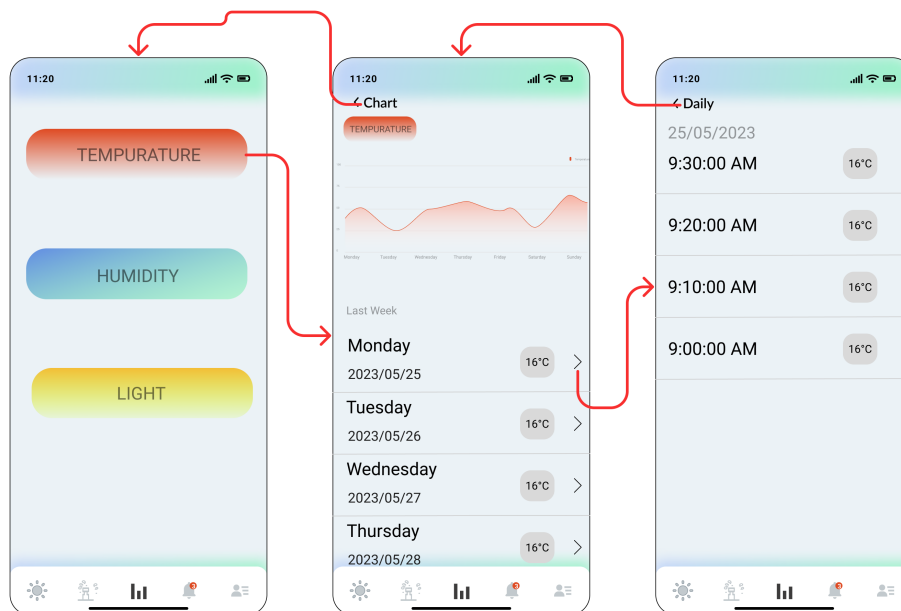
2.2.3 Analyze data

Để xem được thống kê dữ liệu của cây trồng, người dùng cần nhấn vào biểu tượng thứ ba trên thanh điều hướng.

Nhóm chức năng này gồm 3 màn hình chính:

- Màn hình điều hướng: Cho phép người dùng chuyển đến màn hình xem một trong 3 thông số: nhiệt độ, độ ẩm hoặc ánh sáng
- Màn hình tổng quan thông số: Hiển thị biểu đồ giá trị quan trắc theo các mốc thời gian
- Màn hình hiển thị giá trị quan trắc cụ thể trong ngày

Screen Flow cho nhóm chức năng trực hóa quan dữ liệu:



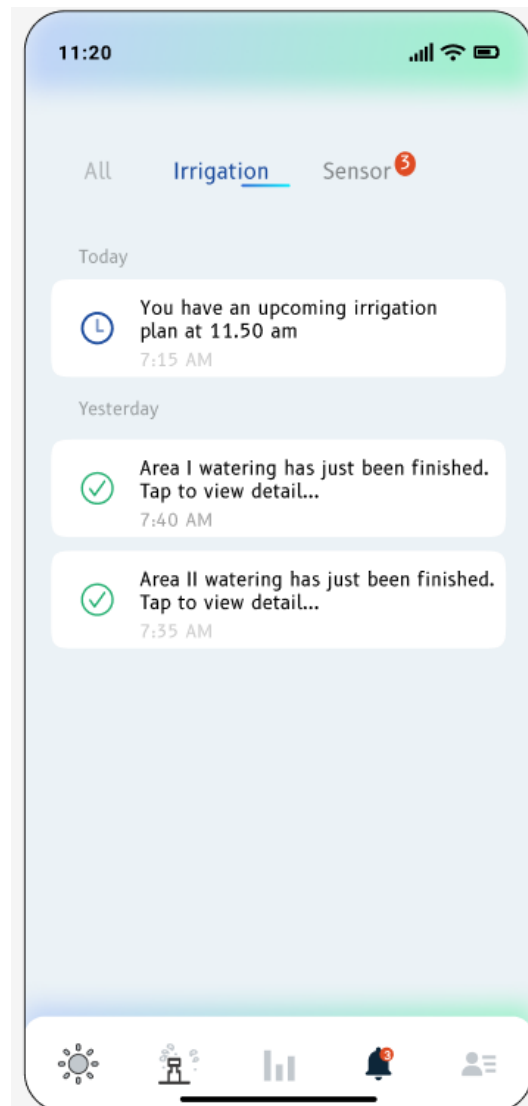
Hình 5: Chart Screen Flow

Các đối tượng quan trọng:

- Nút lựa chọn thông số tại trang điều hướng: Cho phép người dùng chuyển đến trang xem thông số tổng quan của nhiệt độ, độ ẩm hoặc ánh sáng tương ứng với từng thông số.
- Nút xem chi tiết dữ liệu theo ngày tại trang thông tin tổng quan: Cho phép người dùng chuyển đến màn hình xem dữ liệu chi tiết của từng ngày
- Các nút quay về ở đầu mỗi trang: Cho phép người dùng quay về trang trước đó

2.2.4 Notification

Người dùng xem các thông báo bằng cách nhấn vào biểu tượng thứ tư trên thanh điều hướng. Màn hình của trang thông báo:

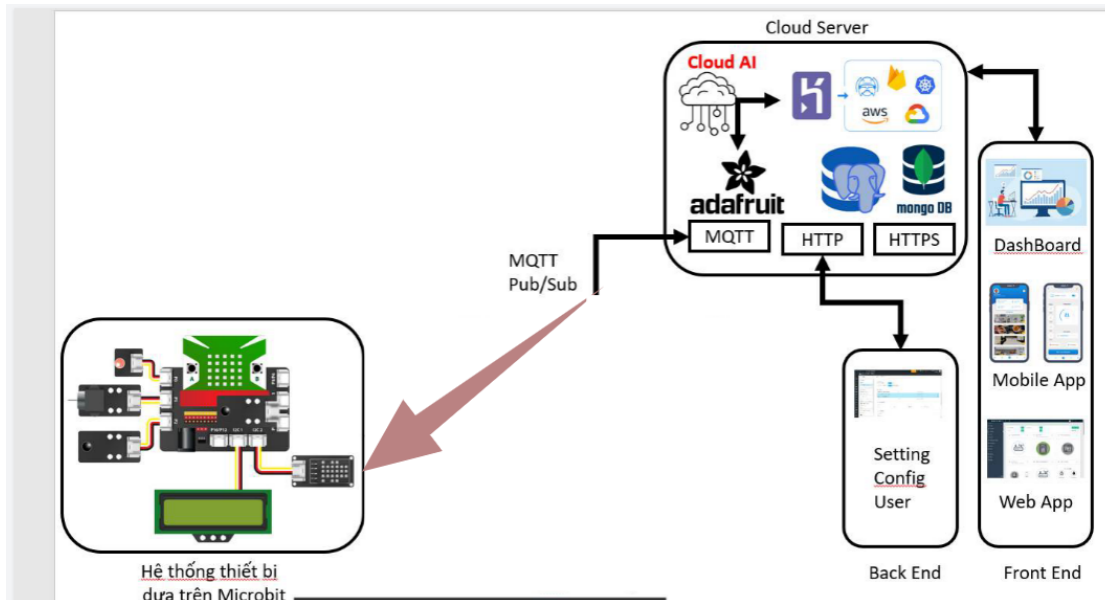


Hình 6: Notification Screen Flow

Người dùng cũng có thể chọn và xóa bớt những thông báo không cần thiết bằng cách nhấn vào thông báo và chọn xóa.

3 Architecture Design

3.1 Hệ thống nhóm đề xuất



Hình 7: Hệ thống đề xuất

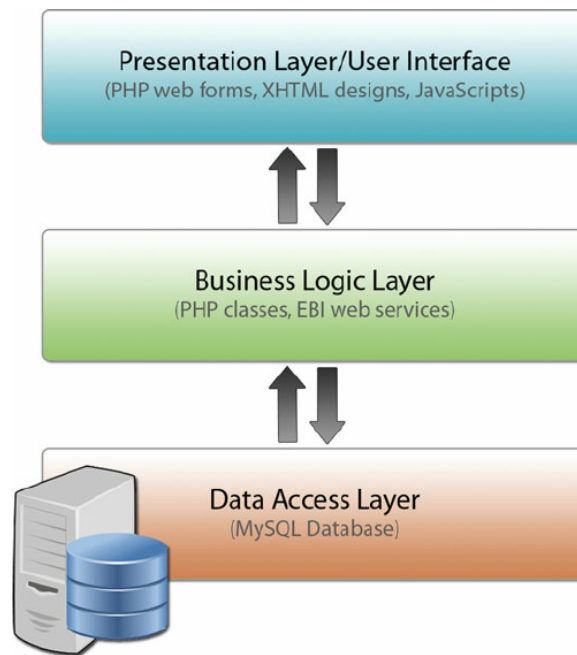
- Nhóm đã lựa chọn bỏ qua thực hiện kết nối Serial đến Gateway từ Hệ thống thiết bị dựa trên Yolo:bit(thay thế Microbit) mà sẽ kết nối thẳng đến CloudSever.
- Hệ thống thiết bị sẽ thực hiện MQTT Pub/Sub để gửi và cập nhật dữ liệu, ở hệ cơ sở dữ liệu nhóm đã chọn MongoDB thực hiện lấy dữ liệu từ MQTT về để thực hiện gửi dữ liệu từ MongoDB đến giao diện hiển thị của hệ thống.
- Khi người dùng có thao tác thay đổi các chức năng dữ liệu sẽ được cập nhật đến Back-End (MongoDB) và sẽ tiếp tục thực hiện gửi dữ liệu đến AdaFruit và truyền thẳng đến hệ thống thiết bị mà ta đang sử dụng và thực hiện thay đổi trạng thái, tín hiệu của thiết bị hiện tại.

3.2 Kiến trúc 3 tầng (Three-layer Architecture)

Mô hình 3-layer gồm có 3 phần chính:

1. **Presentation Layer (GUI)** : Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện (win form, web form,...) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).
2. **Business Logic Layer (BLL)**: Layer này phân ra 2 thành nhiệm vụ :
 - Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.

- Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.
3. **Data Access Layer (DAL):** Lớp này có chức năng giao tiếp với hệ quản trị cơ sở dữ liệu như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa, ...).

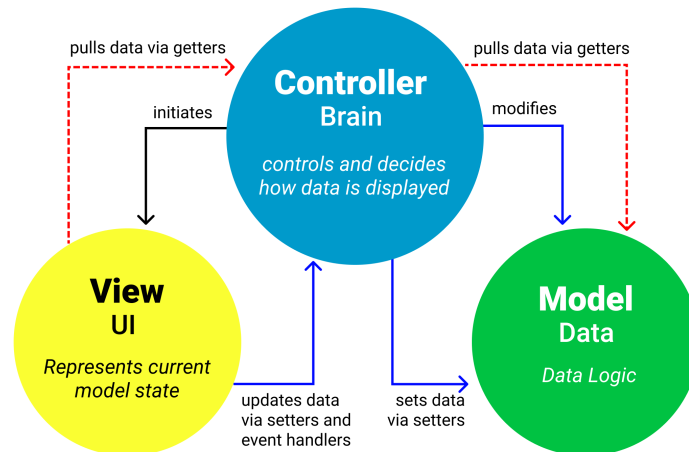


Hình 8: Three-layer Architecture

3.3 Mô hình MVC

MVC (MVC Design Pattern) là viết tắt của Model - View - Controller. Đó là một mẫu kiến trúc, mô hình lập trình phổ biến được sử dụng để tạo cấu trúc cho nhiều trang web, ứng dụng tiên tiến.

MVC Architecture Pattern



Hình 9: MVC Architecture pattern

Mô hình MVC bao gồm ba thành phần thiết yếu:

1. **Model**: Là component trung tâm của kiểu thiết kế này. Nó trực tiếp xử lý các dữ liệu logic và các quy tắc của ứng dụng. Trong ứng dụng này, model có 1 vài chức năng cụ thể như sau: Xác thực login, cập nhật dữ liệu, tạo các thông báo, lưu trữ thông tin, kiểm tra phân công công việc,...
2. **View**: Định nghĩa và quyết định cách mà dữ liệu được hiển thị với người dùng. Cụ thể trong ứng dụng này là hiển thị các form login, trang lịch làm việc, trang thông tin nhân viên,...
3. **Controller**: Xử lý các tương tác của người dùng hay còn gọi là event (ví dụ: bấm nút, click chuột,...), truyền những event đó đến view và model trong tình huống cụ thể của ứng dụng này. Với tình huống cụ thể là ứng dụng này thì những chức năng chính của Controllers là ánh xạ các tương tác của người dùng đến các cập nhật model và chọn view phù hợp để hiển thị.

Tóm lại, ta có thể ví von qua ví dụ sau:

- Controller là Nhạc trưởng.
- Model là các Nhạc công.
- View là bài nhạc được thể hiện của cả dàn nhạc.

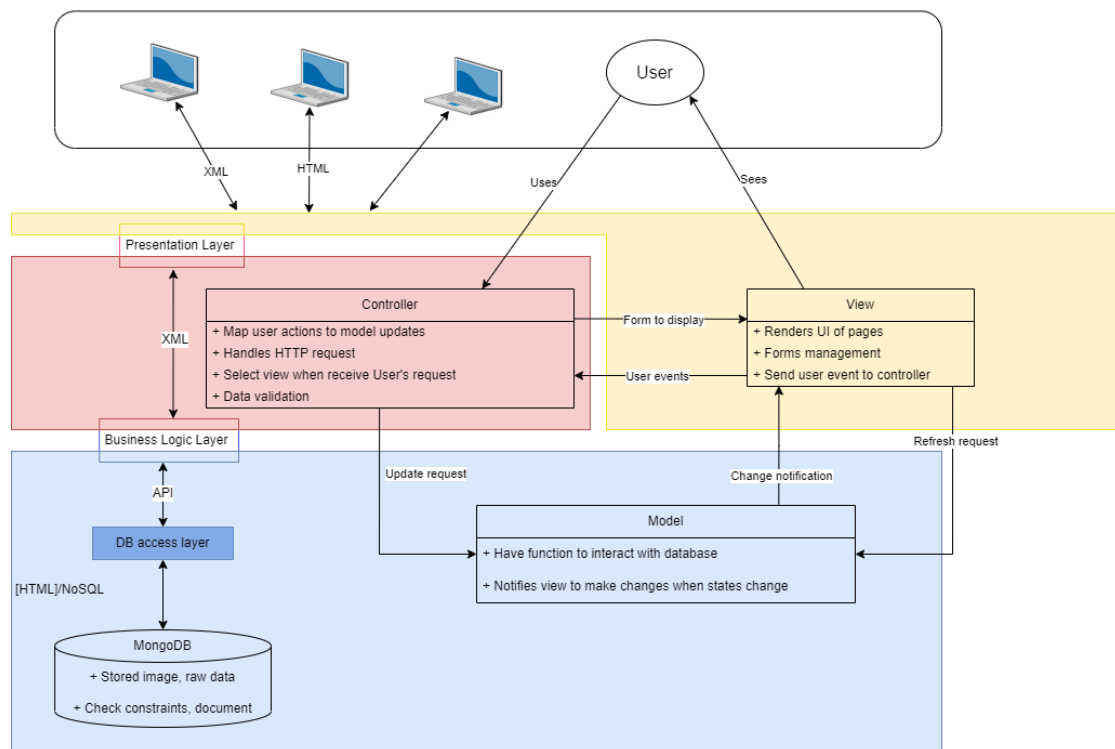
3.4 Lý do nhóm chọn mô hình MVC

Nhóm lựa chọn mô hình MVC vì các lợi ích mà mô hình này đem lại như sau:

- **Tiết kiệm băng thông:** Vì không sử dụng viewstate nên MVC rất nhẹ và tiết kiệm được diện tích của băng thông. Khi cần tương tác gửi và nhận dữ liệu liên tục, người dùng có thể sử dụng các ứng dụng trên web. Điều này giúp website có thể hoạt động ổn định và tốt hơn.
- **Dễ dàng kiểm tra:** Nhờ có MVC ta sẽ chia code thành 2 thành phần front-end và back-end riêng biệt. Do đó, ta sẽ dễ dàng hơn trong việc kiểm tra, rà soát lỗi, đảm bảo được chất lượng và độ uy tín cho phần mềm trước khi tới tay người dùng.
- **Chức năng Separation of Concern:** Cho phép phân tách một cách rõ ràng các thành phần model, data, giao diện hay nghiệp vụ.
- **Tính kết hợp:** Ta có thể thoải mái viết code trên nền tảng web khi tích hợp ở mô hình MVC để giảm tải dữ liệu server.
- **Tính đơn giản:** Kết cấu của mô hình MVC tương đối đơn giản và dễ dàng sử dụng ngay cả khi bạn không có chuyên môn.

3.5 Architectural diagram

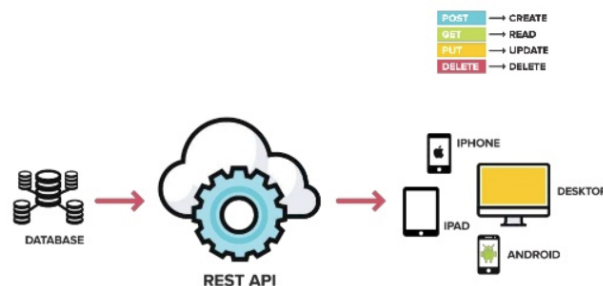
Xem chi tiết tại link này: [Architectural diagram](#)



Hình 10: Architectural diagram

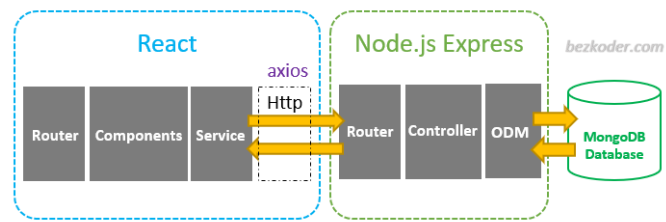
3.6 Kế hoạch dự định hiện thực

1. **Đối với lớp Presentation layer:** Nhóm sẽ sử dụng ReactJS, một thư viện rất phổ biến của Javascript được phát triển bởi Facebook và tạo ra để xây dựng giao diện người dùng có khả năng tương tác tốt và nhanh chóng cho các ứng dụng web và di động. Nó là một thư viện mã nguồn mở, xây dựng dựa trên các component, giao diện người dùng chỉ chịu trách nhiệm cho tầng view của ứng dụng (<https://reactjs.org/>).
2. **Đối với lớp Application layer:** Nhóm dự định sử dụng RESTful API (<https://restfulapi.net/>) để xử lý các yêu cầu và thực thi xử lý nghiệp vụ (business logic). Vì đây là một kiểu kiến trúc được sử dụng rộng rãi và đã được thiết lập tốt để xây dựng các web với các ưu điểm chính như sau:
 - Giúp cho ứng dụng rõ ràng hơn.
 - REST URL đại diện cho resource chứ không phải hành động.
 - Dữ liệu được trả về với nhiều định dạng khác nhau như: xml, html, json.
 - Code đơn giản và ngắn gọn.
 - REST chú trọng vào tài nguyên của hệ thống.



Hình 11: RESTful API

3. **Đối với lớp Data access layer:** Nhóm có thể sử dụng đến MongoDB (<https://www.mongodb.com/docs/>), là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở đa nền tảng viết bằng C++. Các lý do chính mà nhóm quyết định sử dụng MongoDB là:
 - Hệ sinh thái: ReactJS + NodeJS + MongoDB
 - Schema linh hoạt: Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau.
 - Cấu trúc đối tượng rõ ràng: Tuy rằng cấu trúc của dữ liệu là linh hoạt nhưng đối tượng của nó được xác định rất rõ ràng. Sử dụng bộ nhớ nội tại, nên truy vấn sẽ rất nhanh.
 - MongoDB rất dễ mở rộng.
 - Không có các join: Điều này cũng góp phần tạo nên tốc độ truy vấn cực nhanh trên MongoDB.
 - MongoDB phù hợp cho các ứng dụng Realtime.

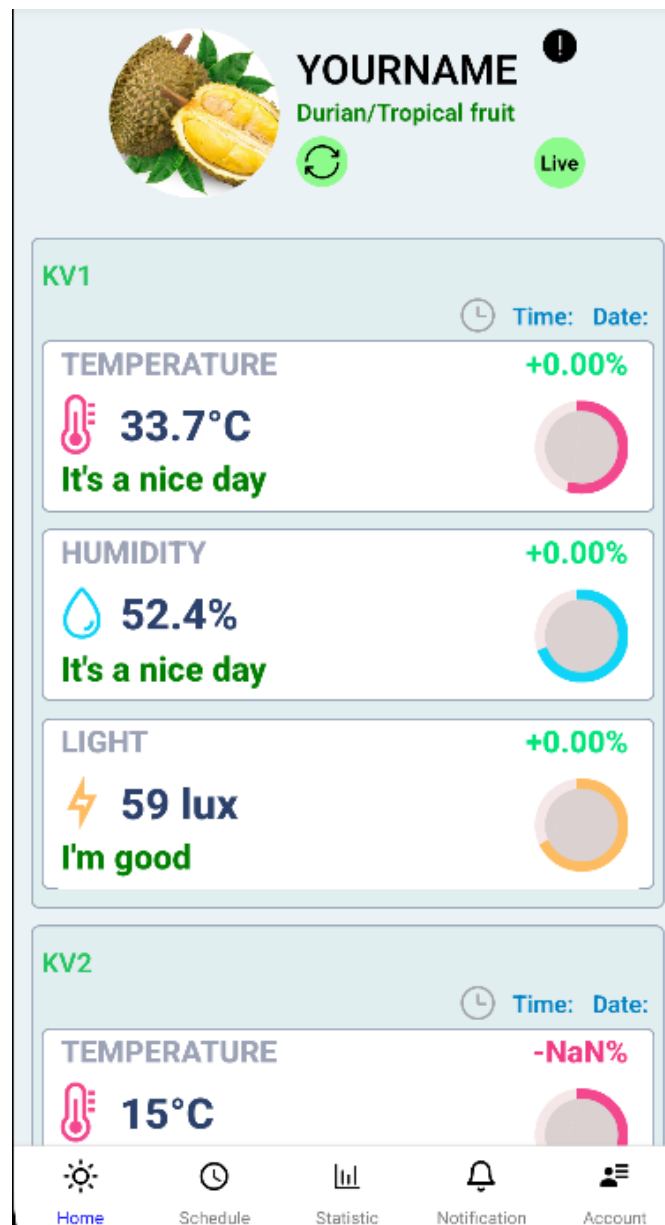


Hình 12: MongoDB

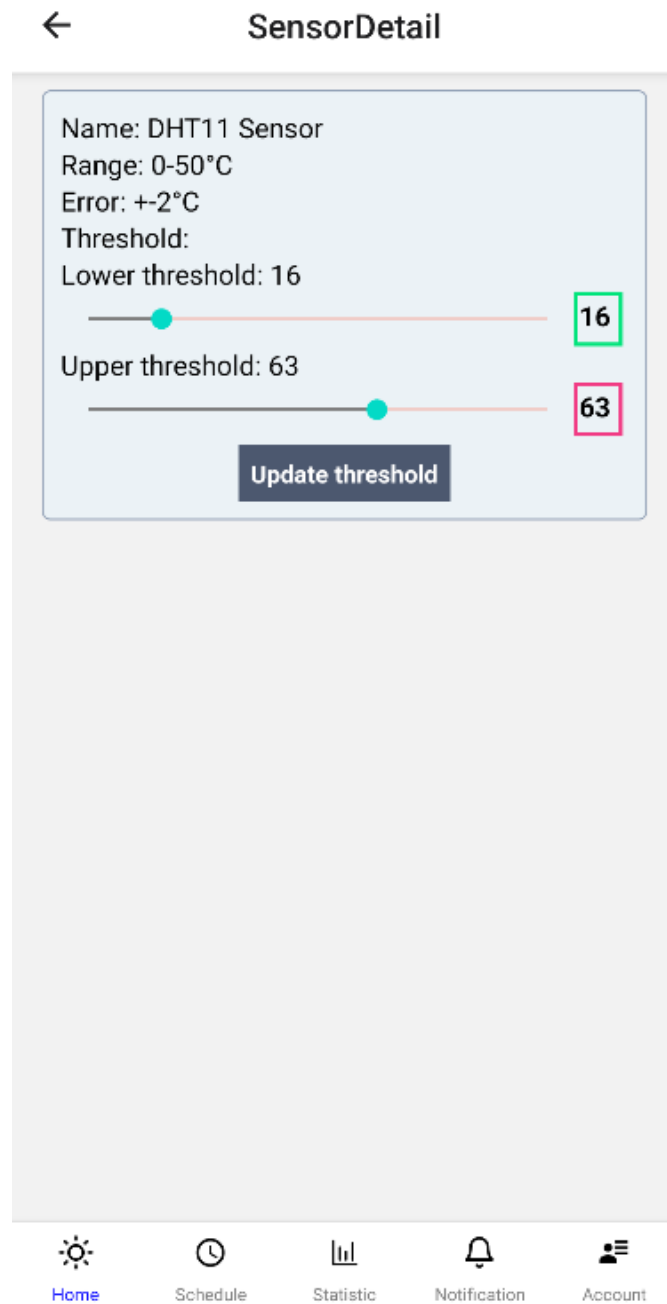
4 Implementation - Sprint

4.1 Giao diện ứng dụng

4.1.1 Giao diện phân tích trạng thái của môi trường và quản lý các cảm biến

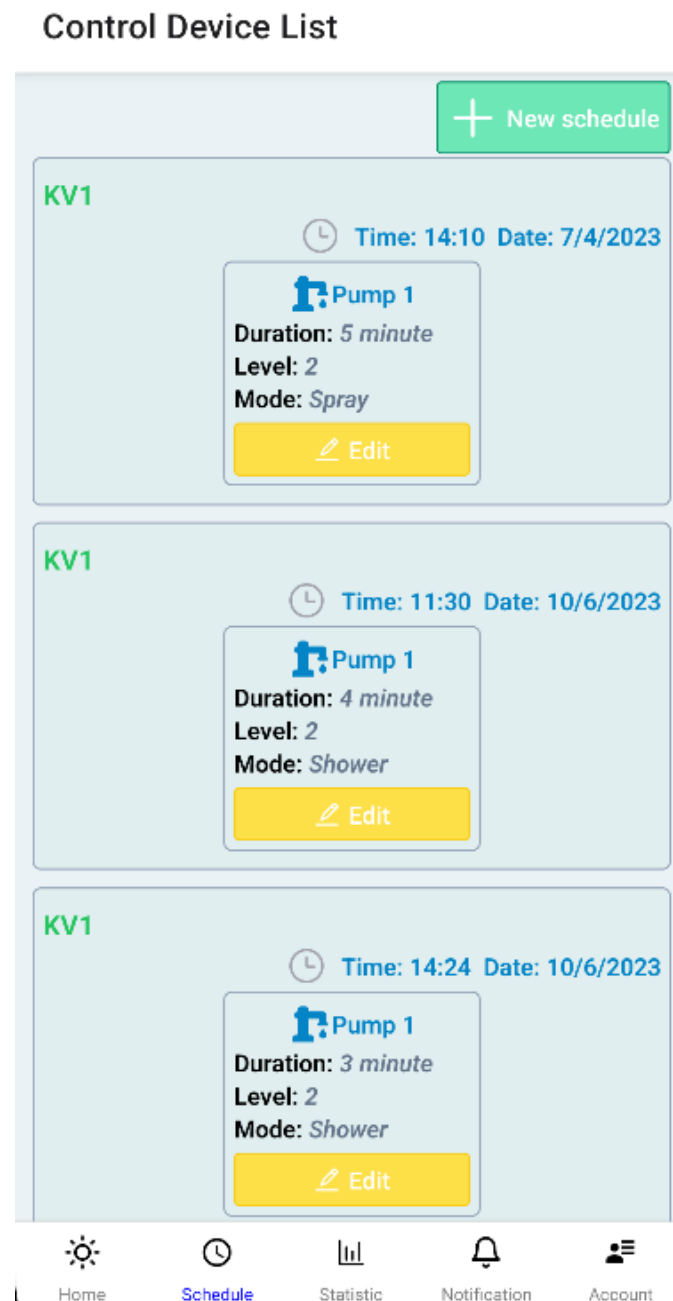


Hình 13: Giao diện Trang chủ phân tích trạng thái môi trường



Hình 14: Giao diện xem chi tiết và quản lý các cảm biến

4.1.2 Giao diện quản lý tưới tiêu



Hình 15: Giao diện lập lịch tưới tiêu

←

Edit Control Device

Pump 1

Duration

Duration minute

Level

☐ 1

☒ 2

☐ 3

Mode


☒ Spray


☐ Shower


☐ Heavy Shower


Active device


Save

 Home

 Schedule

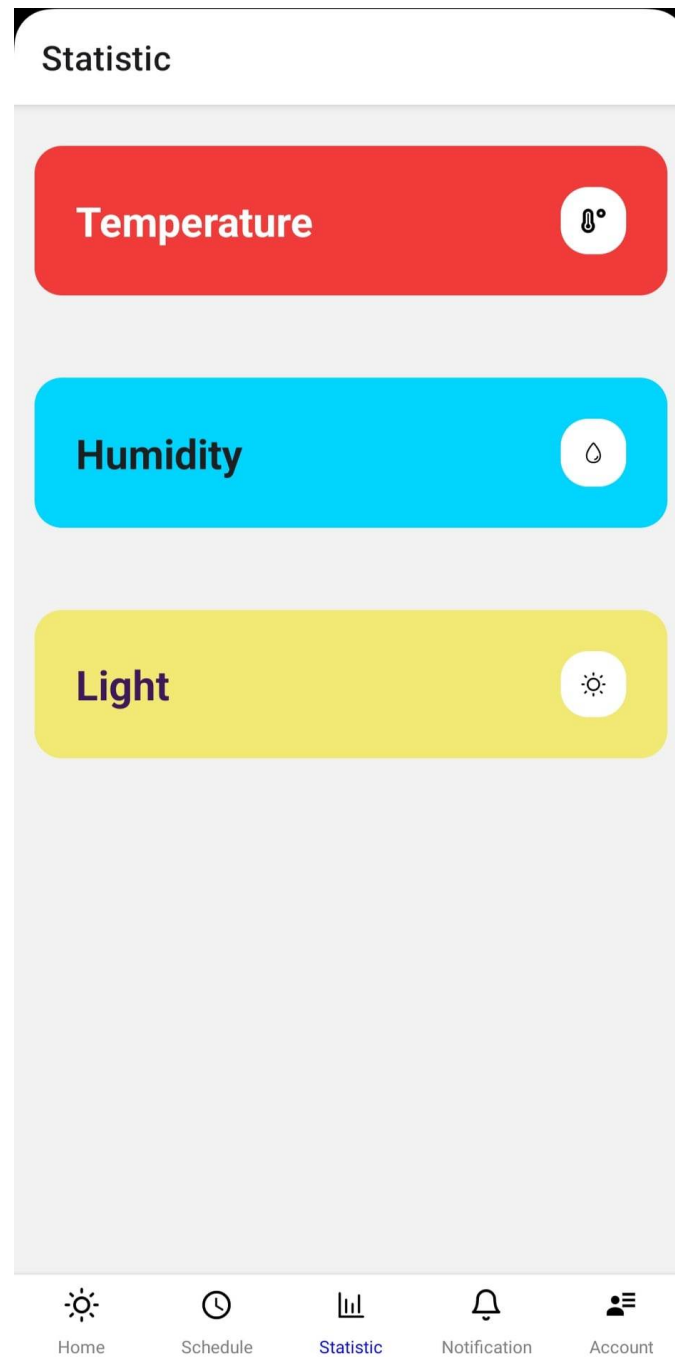
 Statistic

 Notification

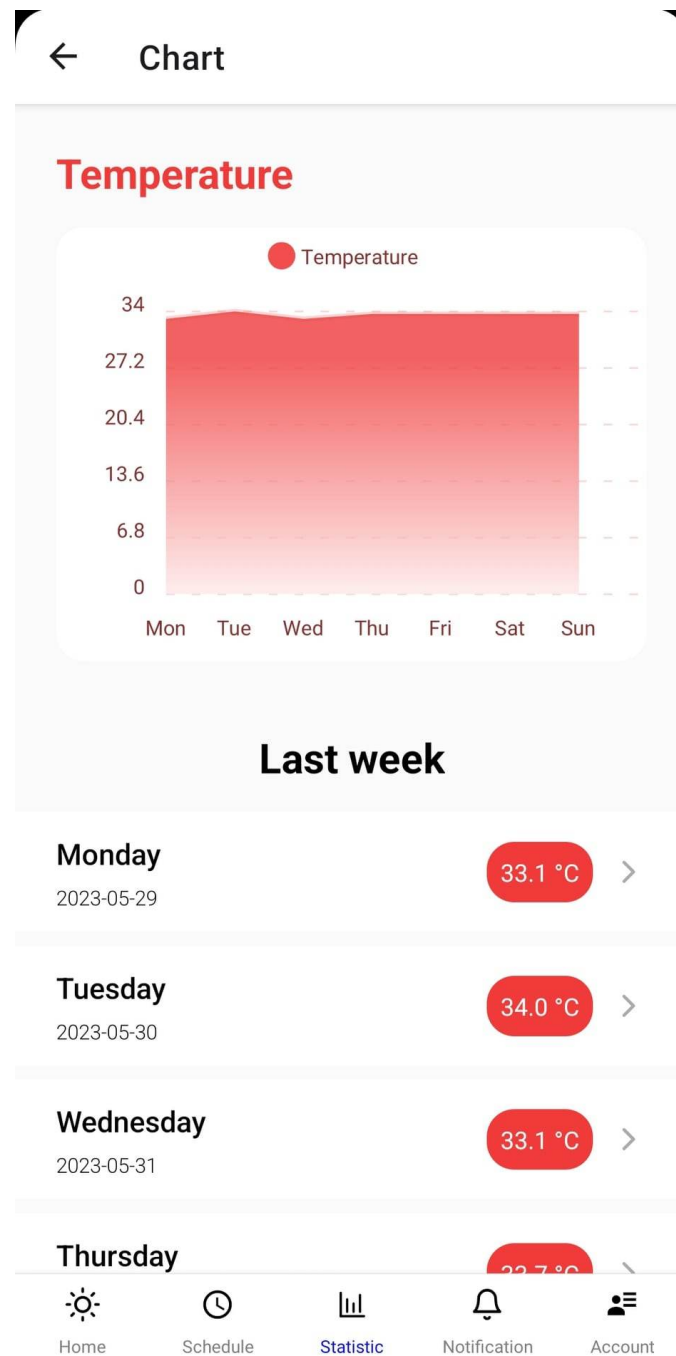
 Account

Hình 16: Giao diện điều chỉnh chế độ tưới nước

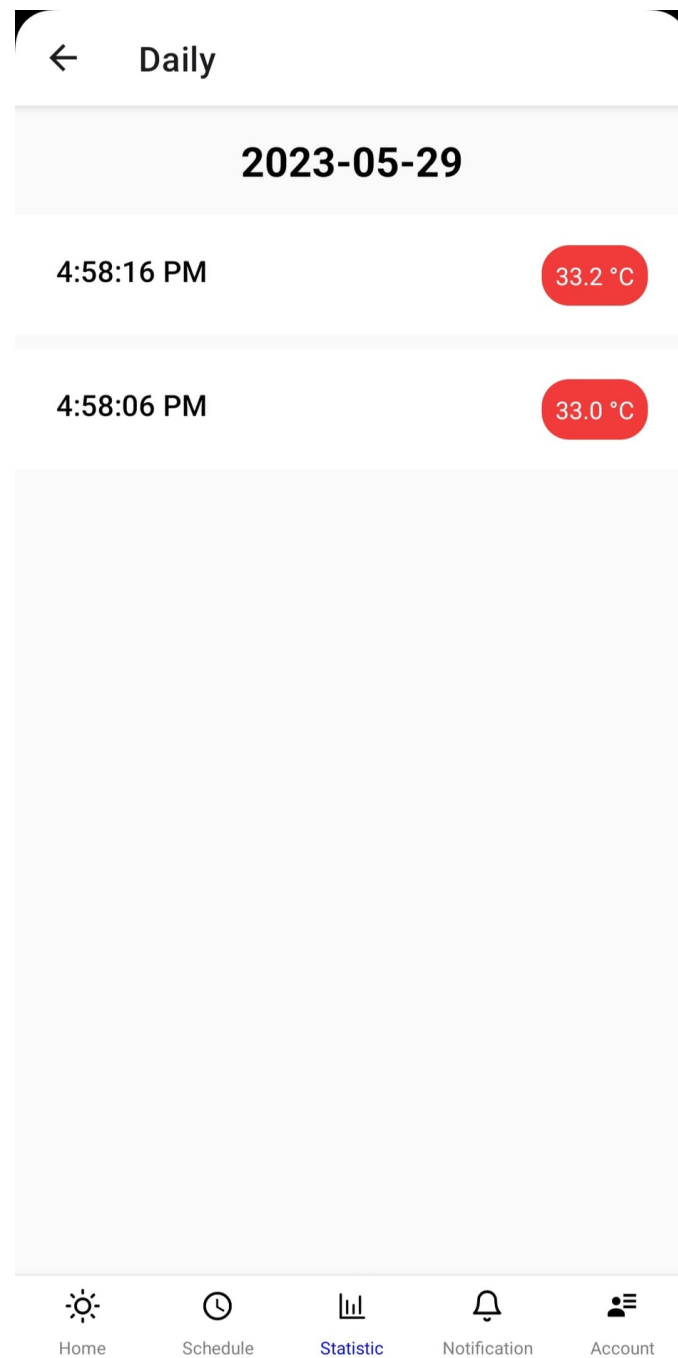
4.1.3 Giao diện phân tích dữ liệu của hệ thống



Hình 17: Giao diện các thông số



Hình 18: Giao diện chi tiết thông số của cây trồng



Hình 19: Giao diện theo ngày của thông số

4.2 Kế hoạch thuyết trình

Nhóm sẽ thuyết trình trên 4 nội dung chính.

1. Về nhóm và dự án

- Giới thiệu về thành viên nhóm, vai trò của từng người
 - Giới thiệu về dự án, các tính năng như phân tích trạng thái của hệ thống, quản lý cảm biến, quản lý tưới tiêu, ...
2. Kết quả của dự án
- Use-case diagram
 - Sản phẩm thiết kế
3. Các công nghệ sử dụng và lý do chọn các công nghệ đó:
- Kiến trúc hệ thống
 - Về công nghệ lập trình, thiết kế. Nhóm sử dụng hệ sinh thái MERN (MongoDB, ExpressJS, ReactJS, NodeJS)
4. Các vấn đề, thách thức, giải pháp khi phát triển phần mềm và bài học rút ra cho nhóm và từng cá nhân.
- Trình bày về rủi ro và thách thức:
 - Về nhóm: Các vấn đề liên quan tới xung đột ý kiến, khác biệt về kinh nghiệm và kỹ năng từ các thành viên.
 - Về các tài nguyên: Hầu hết làm việc online.
 - Về kinh nghiệm: Chưa có workflow rõ ràng, ít kinh nghiệm về thiết kế phần mềm, ít hiểu biết về trải nghiệm người dùng và sự đánh giá khách quan lẫn chủ quan của yếu tố môi trường.
 - Giải pháp sử dụng để giải quyết các vấn đề đó.
 - Về nhóm: Học hỏi giữa các thành viên, luôn đặt ra thách thức ngay và luôn, chia sẻ những khóa học về các công nghệ sử dụng.
 - Về tài nguyên: Luôn ghi lại những buổi họp nhóm để các thành viên xem lại
 - Về kinh nghiệm : Mỗi thành viên đều tự học những kiến thức mới, sử dụng Git để quản lý dự án và tiến độ công việc
 - Bài học dành cho nhóm và từng thành viên
 - Về kiến thức: Quản lý dự án phần mềm, quá trình thiết kế phần mềm, các ngôn ngữ lập trình ứng dụng (app), hiểu thêm về kết nối từ việc thu thập dữ liệu đến hiển thị trực quan.
 - Về kỹ năng mềm: làm việc nhóm, trao đổi giữa các thành viên, quản lý thời gian và công việc.



References

- [1] Nguyen Hoang Phu Thinh, *Use Case Diagram và 5 sai lầm thường gặp*.
- [2] What is MongoDB ?, *What is MongoDB?*.
- [3] How to connect MongoDB with AdaFruit?, *Integrate the Adafruit IO API with the MongoDB API*.
- [4] Design Pattern là gì?, *Design pattern là gì? Tại sao nên sử dụng Design pattern?*.