**Specification Document**

**1. บทนำ (Introduction)**

**1.1 ภาพรวมโครงการ FarmIQ™**

FarmIQ™ คือแพลตฟอร์ม Smart Farming ที่รวม IoT, AI, และระบบอัตโนมัติ เพื่อนำข้อมูลมาแปลงเป็นการตัดสินใจแบบ Real-time โดยมุ่งเน้นลดต้นทุน เพิ่มประสิทธิภาพ และลดความผิดพลาดจากมนุษย์ สร้างฐานข้อมูลสำหรับการต่อยอดสู่ระบบอัจฉริยะที่รองรับการขยายตัวในอนาคต

**1.2 ประโยชน์ที่ฟาร์มจะได้รับจากการใช้ระบบ Dashboard**

* ลดต้นทุนแรงงานและความผิดพลาดของมนุษย์
* เพิ่มประสิทธิภาพในการจัดการข้อมูล และวางแผนฟาร์ม
* สนับสนุนการตัดสินใจที่รวดเร็วและแม่นยำด้วยข้อมูลที่เชื่อถือได้

**1.3 วัตถุประสงค์และเป้าหมายของ Dashboard**

* แสดงข้อมูลและสถานะต่างๆ ของฟาร์มอย่างชัดเจนและ Real-time
* รองรับการแจ้งเตือนเมื่อมีเหตุการณ์ผิดปกติ
* ช่วยให้ผู้ใช้งานสามารถควบคุมอุปกรณ์ในฟาร์มผ่าน Dashboard
* สนับสนุนการสร้างรายงานอัตโนมัติ และการวิเคราะห์ข้อมูลเพื่อการตัดสินใจ
* การเพิ่มประสิทธิภาพการใช้ทรัพยากรในฟาร์ม (อาหาร น้ำ พลังงาน)
* การเพิ่มขีดความสามารถในการตอบสนองต่อเหตุการณ์ผิดปกติได้เร็วขึ้น
* การสร้างฐานข้อมูลที่เป็นมาตรฐานเพื่อต่อยอด AI/ML ในอนาคต

**2. ขอบเขตงาน (Scope of Work)**

**2.1 ประเภทของ Dashboard**

* Real-time Monitoring Dashboard
* Batch Reporting Dashboard
* Alert & Notification Center

**2.2 ขอบเขตการพัฒนา**

* Frontend: React.js + MUI
* Backend: RESTful APIs (Node.js/FastAPI) -- Optional
* Data Integration: MQTT, InfluxDB/PostgreSQL, AI Model APIs, External APIs (Weather, Market Data)
* การเชื่อมต่อระบบ Dashboard กับ AI/ML Engine เพื่อการวิเคราะห์และแสดงผลข้อมูลเชิงลึก
* การเชื่อมต่อกับฐานข้อมูล InfluxDB/PostgreSQL เพื่อการจัดเก็บข้อมูลแบบ Time-Series อย่างมีประสิทธิภาพ

**3. ผู้ใช้งานและบทบาท (User Roles)**

* **Admin**: สามารถตั้งค่าระบบ จัดการสิทธิ์ผู้ใช้งาน และดูแลระบบ Dashboard ได้ทั้งหมด
* **Operator / Farm Manager**: สามารถสั่งการอุปกรณ์และแก้ไขปัญหาเบื้องต้นผ่านระบบ Dashboard
* **Executive**: สามารถดูภาพรวมผลการดำเนินงานและวิเคราะห์ข้อมูลเชิงกลยุทธ์เพื่อการตัดสินใจระดับสูง

**4. Functional Requirements**

**4.1 Data Ingestion & Pipeline**

* บันทึกข้อมูล Time-Series ลง InfluxDB พร้อมจัดการ Buffering, Retention, Downsampling
* ตรวจสอบคุณภาพข้อมูล (Data Validation, Outlier Detection) ก่อนเก็บ

**4.2 Data Processing & Analytics**

* คำนวณค่า FCR, ADG, Daily Gain, Cpk/Ppk อัตโนมัติ
* รันโมเดล AI (Forecasting, Anomaly Detection, Predictive Maintenance) ผ่าน API
* จัดเก็บผลลัพธ์และ Log การคำนวณ

**4.3 API Layer**

* RESTful/GraphQL API สำหรับดึงข้อมูล Raw และ Processed
* รองรับ Query แบบ Filter (Time Range, Farm, Sensor Type) และ Pagination
* Security: OAuth2, Token-based, Rate Limiting

**4.4 Data Visualization**

* Charts: Time-Series Trends (Weight, Temp, Humidity, FCR/ADG) Real-time & Historical
* Comparative Analysis: เปรียบเทียบระหว่างโรงเรือนหรือฟาร์ม
* Status Table: สถานะ Sensor/Actuator (Online/Offline, Last Seen)
* Threshold & Alerts: ตั้งค่าเกณฑ์แจ้งเตือน, Highlight กราฟเมื่อเกิน Threshold
* Real-time Refresh: อัปเดตข้อมูลทุก 1-5 วินาที พร้อม Manual Refresh

**4.5 Control & Interaction**

* Command Buttons: Feed, Light, Environment Control ผ่าน Dashboard
* Preset Management: บันทึก/เรียกใช้ Profile การตั้งค่าอุปกรณ์ (Preset)
* Search & Filter: ช่วงเวลา, ฟาร์ม, โรงเรือน, Sensor Type
* Activity Logs: บันทึกประวัติการสั่งงานและการเปลี่ยนแปลงค่า Setting
* User Preferences: บันทึกการตั้งค่ามุมมอง Dashboard แบบเฉพาะผู้ใช้ (Custom View)

**4.6 Reporting & Export**

* Scheduled Reports: Daily/Weekly/Monthly พร้อมการแจ้งเตือนทาง Email/Slack
* Export: PDF, Excel, CSV โดยเลือก Fields และ Period ได้เอง
* Customizable Templates: กำหนด Layout และ Metrics ในรายงานได้
* On-demand Report Generation: ผู้ใช้สามารถสร้างรายงานตามต้องการ

**4.7 Notification & Alerting**

* Multi-channel Alerts: ส่งผ่าน Email, SMS, Push Notification
* Alert Management: Acknowledge, Snooze, Escalation Rules
* Alert Dashboard: แสดงสถานะ Alert ทั้งหมดพร้อม Filters

**4.8 User Management & Security**

* Authentication: SSO (SAML/OAuth2), 2FA Option
* Authorization: Role-Based Access Control (RBAC), Multi-Tenancy Support
* Audit Trail: บันทึกกิจกรรมสำคัญ (Login, Data Export, Setting Change)
* Password Policy & Session Management: Expiry, Idle Timeout

**4.9 Administration & Configuration**

* System Settings: Configure Data Retention, Thresholds, Report Schedule
* Health Monitoring: System Status, Data Pipeline Health, API Uptime
* Maintenance Mode: ปิดการใช้งานชั่วคราวพร้อมแจ้งผู้ใช้งาน

**5. Non-Functional Requirements**

**5.1 UI/UX Guidelines**

* ดีไซน์ทันสมัย สะอาดตา และเน้น Corporate Identity
* รองรับ Responsive Design (Desktop/Tablet/Mobile)
* การใช้งานแบบ Minimal Click (ลดจำนวนขั้นตอนหรือคลิกเพื่อเข้าถึงข้อมูลสำคัญ)

**5.2 Performance**

* เวลาในการโหลดหน้า Dashboard ไม่เกิน 2 วินาที
* รองรับผู้ใช้งานพร้อมกันไม่น้อยกว่า 100 คน
* ทดสอบและแสดงผลการทดสอบการทำงานของระบบภายใต้สภาวะโหลดสูงสุด

**5.3 Scalability & Maintainability**

* ออกแบบเพื่อรองรับการขยายฟีเจอร์ในอนาคต
* จัดทำเอกสารการติดตั้งและ Deployment อย่างละเอียด
* จัดทำคู่มือการบำรุงรักษา และคู่มือการแก้ไขปัญหาสำหรับทีม IT ขององค์กร

**5.4 Security & Compliance**

* เข้ารหัสข้อมูลด้วย TLS
* ปฏิบัติตามมาตรฐาน PDPA
* มีระบบแจ้งเตือนอัตโนมัติกรณีระบบล่มหรือทำงานผิดปกติ (System Health Alert)

**5.5 Availability & Reliability**

* ระบบต้องมี Uptime ไม่น้อยกว่า 99.5%
* มีระบบสำรอง (Backup/Failover)

**6. การเชื่อมต่อและข้อมูล (Integration & Data Sources)**

* เรียกข้อมูลจาก AI/Analytics Layer ผ่าน API
* ดึงข้อมูลจาก External API เช่น Weather API, Market Data

**7. Deliverables**

* Mockups / Wireframes
* Source Code พร้อม README และคำอธิบายชัดเจน
* API Reference Document
* User Manual และ Training Guide

**8. Timeline & Milestones**

* Kickoff Meeting (สัปดาห์ที่ 1)
* Design Review (สัปดาห์ที่ 3)
* Development (สัปดาห์ที่ 4-12)
* User Acceptance Testing (สัปดาห์ที่ 13-14)
* Go-Live (สัปดาห์ที่ 15)

**9. Acceptance Criteria**

* ระบบทำงานถูกต้องตามฟังก์ชันที่ระบุ
* ระดับ Bug Tolerance ไม่เกิน 5% ในช่วง UAT
* ผ่าน Performance Benchmark (Response Time, Uptime)

**10. ข้อกำหนดอื่น ๆ (Optional)**

* สัญญา Support & Maintenance 12 เดือนหลังส่งมอบ
* Warranty Period 6 เดือนหลังจาก Go-Live
* การอบรมและถ่ายทอดความรู้ (Training & Knowledge Transfer)
  + การอบรมการใช้งานระบบให้กับผู้ใช้งาน (Admin, Operator)
  + การอบรมการดูแลรักษาและแก้ไขปัญหาเบื้องต้นให้ทีม IT หรือ Support ขององค์กร

**11. เอกสารประกอบเพิ่มเติม (Documentation)**

* Technical Documentation (System Architecture, Database Design)
* Troubleshooting Guide

**12. การส่งมอบ Source Code และ Version Control**

* ระบุการใช้งานระบบ Version Control (Git) และการจัดการ Repository

**13. SLA (Service Level Agreement)**

* ระยะเวลาในการตอบสนองเมื่อเกิดปัญหา (Response Time)
* ระยะเวลาในการแก้ไขปัญหา (Resolution Time)
* ข้อตกลงในการอัปเดตระบบและแก้ไข Bug หลัง Warranty Period หมดอายุ
* รูปแบบการคิดค่าใช้จ่ายในการ Support & Updates

**---------------- English version -------------------**

**Specification Document**

**1. Introduction**

**1.1 Project Overview: FarmIQ™**

FarmIQ™ is a Smart Farming platform that integrates IoT, AI, and automation to transform data into real-time decision-making. It focuses on reducing costs, improving efficiency, and minimizing human error, while establishing a foundation for future expansion into an intelligent, scalable system.

**1.2 Objectives and Goals of the Dashboard**

* Present farm data and status clearly and in real time
* Support alerting for abnormal events
* Enable users to control on-site equipment via the dashboard
* Facilitate automated report generation and data-driven analysis
* Optimize resource utilization (feed, water, energy)
* Improve response time to incidents
* Provide standardized data for future AI/ML enhancements

**2. Scope of Work**

**2.1 Dashboard Types**

* Real-time Monitoring Dashboard
* Batch Reporting Dashboard
* Alerts & Notification Center

**2.2 Development Scope**

* **Frontend:** React.js with MUI
* **Backend:** RESTful APIs (Node.js or FastAPI) -- Optional
* **Data Integration:** MQTT, InfluxDB/PostgreSQL, AI Model APIs, External APIs (Weather, Market Data)
* **AI/ML Integration:** Connect dashboard to analytics engine for advanced insights
* **Time-Series Storage:** Efficient data storage in InfluxDB/PostgreSQL with retention and downsampling

**3. User Roles**

* **Admin:** Full system configuration, user management, and dashboard oversight
* **Operator / Farm Manager:** Monitor status and issue control commands; perform basic troubleshooting
* **Executive:** View high-level performance summaries and strategic KPIs

**4. Functional Requirements**

**4.1 Data Ingestion & Pipeline**

* Store time-series data in InfluxDB/PostgreSQL with buffering, retention policies, and downsampling
* Perform data validation and outlier detection prior to storage

**4.2 Data Processing & Analytics**

* Automatically compute metrics like FCR, ADG, Daily Gain, Cpk/Ppk
* Execute AI models for forecasting, anomaly detection, and predictive maintenance via API
* Log and store computation results and model outputs

**4.3 API Layer**

* Provide RESTful/GraphQL APIs for raw and processed data access
* Support filtering (time range, farm, sensor type), sorting, and pagination
* Implement security measures: OAuth2, token-based authentication, and rate-limiting

**4.4 Data Visualization**

* **Time-Series Charts:** Real-time and historical trends for weight, temperature, humidity, FCR/ADG
* **Comparative Analysis:** Side-by-side comparisons across farms or houses
* **Status Table:** Sensor/actuator health (online/offline, last seen)
* **Threshold & Alerts:** Configurable alert thresholds; visual highlighting when crossed
* **Auto Refresh:** Data auto-updates every 1–5 seconds, with manual refresh option

**4.5 Control & Interaction**

* **Command Controls:** Buttons for feed, lighting, and environmental adjustments
* **Preset Management:** Save and recall device configuration profiles
* **Search & Filter:** Filter data by date range, farm, house, sensor type
* **Activity Log:** Track control actions and configuration changes
* **User Preferences:** Personalized dashboard views and settings

**4.6 Reporting & Export**

* **Scheduled Reports:** Automated daily, weekly, and monthly reports with email/Slack notifications
* **Export Options:** PDF, Excel, CSV exports with selectable fields and date ranges
* **Custom Templates:** Define report layouts and metrics
* **On-Demand Reports:** Generate ad-hoc reports as needed

**4.7 Notification & Alerting**

* **Channels:** Email, SMS, and push notifications
* **Alert Management:** Acknowledge, snooze, and set escalation rules
* **Alert Dashboard:** Central view of active and historical alerts with filtering

**4.8 User Management & Security**

* **Authentication:** SSO (SAML/OAuth2) with optional 2FA
* **Authorization:** Role-based access control (RBAC) and multi-tenancy support
* **Audit Trail:** Record critical actions such as logins, exports, and configuration edits
* **Password & Session Policies:** Enforce strong passwords, session timeouts, and expiry

**4.9 Administration & Configuration**

* **System Settings:** Configure data retention, alert thresholds, and report schedules
* **Health Monitoring:** Dashboard for system status, data pipeline health, and API uptime
* **Maintenance Mode:** Temporarily disable user access with notification

**5. Non-Functional Requirements**

**5.1 UI/UX Guidelines**

* Modern, clean design aligned with corporate branding
* Responsive layout for desktop, tablet, and mobile devices
* Minimal-click workflows for core tasks

**5.2 Performance**

* Page load times under 2 seconds
* Support at least 100 concurrent users
* Conduct load testing and publish results

**5.3 Scalability & Maintainability**

* Modular architecture to facilitate future feature expansion
* Comprehensive installation, deployment, and maintenance documentation
* Troubleshooting guide for IT teams

**5.4 Security & Compliance**

* Encrypt data in transit with TLS
* Comply with data protection regulations (e.g., PDPA)
* Implement system health alerts for outages or failures

**5.5 Availability & Reliability**

* Guarantee 99.5% uptime
* Provide backup and failover mechanisms

**6. Integration & Data Sources**

* AI/Analytics results via secure APIs
* External sources (weather, market pricing) via RESTful APIs

**7. Deliverables**

* High-fidelity mockups and wireframes
* Source code repository with README and inline documentation
* API reference documentation
* User manual and training materials

**8. Timeline & Milestones**

* **Week 1:** Project kickoff
* **Week 3:** Design review
* **Weeks 4–12:** Development phase
* **Weeks 13–14:** User acceptance testing
* **Week 15:** Go-live

**9. Acceptance Criteria**

* All functional requirements implemented and verified
* Bug rate under 5% during UAT
* Meet or exceed performance benchmarks (response times, uptime)

**10. Optional Requirements**

* **12-month Support & Maintenance Contract** post-delivery
* **6-month Warranty** after go-live
* **Training & Knowledge Transfer:**
  + User training for Admins and Operators
  + IT training for basic support and troubleshooting

**11. Additional Documentation**

* Technical documentation (architecture diagrams, database schema)
* Troubleshooting guide

**12. Source Code Delivery & Version Control**

* Git-based repository management and branching strategy

**13. Service Level Agreement (SLA)**

* **Response Time:** Defined maximum time to acknowledge incidents
* **Resolution Time:** Defined maximum time to resolve incidents
* **Post-Warranty Support:** Terms and rates for bug fixes and updates after warranty period