POMI
PAITON OPERATION & MAINTENANCE INDONESIA

Business Process Improvement Division

REQUIREMENTS SPECIFICATION FOR NEW SOFTWARE WA-INTEGRATION

Additional information

1. INTRODUCTION

1.1. Company information

PT. POMI is operator of 2,035 MW (contracted) power plant at Paiton – East Java, Indonesia. Unit 3, the first highly efficient 815 MW coal-fired super critical power plant in Indonesia, started commercial operation on 27 March 2012

1.2. Project introduction

The WA-Integration project is a data integration platform that connects various industrial data sources such as AVEVA/PI, MySQL, Oracle, and AI systems with WhatsApp services. It provides a web-based interface for managing data connections and creating triggers based on connected datasets. Users can execute these triggers through a WhatsApp bot, which retrieves data according to the specified connection and trigger, then returns the output as a WhatsApp message. The system consists of three main components — a backend for managing data and triggers, a WhatsApp bot for user interaction, and a web frontend for configuration — enabling users to access and monitor industrial and AI data in real time through a simple and familiar communication platform.

1.3. Glossary

- Node.js: Runtime JavaScript untuk menjalankan backend server.

- Express: Framework web untuk Node.js yang menangani rute dan middleware.

- Next.js: Framework React untuk membangun antarmuka web (frontend).

- React: Library JavaScript untuk membangun tampilan pengguna.

- AVEVA/PI: Sistem historian industri yang menyimpan data proses dan alarm.

- SQLite: Database ringan yang digunakan untuk penyimpanan lokal pada sistem (via `better-sqlite3`).

- Oracle, MySQL: Contoh database relasional yang dapat dihubungkan melalui plugin/driver.

- Data Source: Sumber data apa pun yang terhubung ke sistem (database, historian, atau layanan AI).

- Plugin Driver: Modul yang menambahkan kemampuan koneksi ke jenis data source tertentu.

- whatsapp-web.js: Library untuk mengotomatisasi sesi WhatsApp Web dari server.

- Puppeteer: Library untuk menjalankan dan mengontrol browser headless yang digunakan oleh bot.

- QR Code: Kode yang dipindai untuk mengautentikasi sesi WhatsApp Web.

- Session Files: File yang menyimpan sesi agar tidak perlu scan QR setiap kali bot dijalankan.

- Trigger: Sebuah aturan atau konfigurasi yang, ketika dipanggil atau kondisinya terpenuhi, akan mengeksekusi aksi (mis. menjalankan query dan mengirimkan hasilnya ke user via WhatsApp).

- Trigger Engine: Modul di backend yang mengelola pendaftaran, penyimpanan, dan eksekusi trigger.

- Trigger Groups: Pengelompokan trigger berdasar kategori atau fungsi.

- Query Trigger: Trigger yang menjalankan query ke data source dan mengembalikan hasil.

- REST API: Antarmuka HTTP yang digunakan frontend dan bot untuk berkomunikasi dengan backend.

- Endpoint: URL spesifik di API yang menjalankan fungsi tertentu (mis. `/api/triggers`).

- Middleware: Lapisan fungsi yang dijalankan sebelum handler utama (contoh: autentikasi, logging).

- Route: Rangkaian path dan handler yang menangani permintaan tertentu.

2. PROJECT GOALS

This section provides generic information about the project.

2.1. Aims of the project

Project Objectives

- Integrate monitoring data from multiple sources (AVEVA/PI, Oracle, MySQL, SQLite, and AI connections) into a centralized platform.
- Provide a trigger mechanism that can be defined or added via the web interface and executed by users through the WhatsApp bot.
- Develop a reliable WhatsApp bot capable of sending notifications and responding to data or query requests from operators.
- Offer a web interface (built with Next.js) for managing connections, creating triggers, and viewing history or logs.
- Implement a modular architecture (plugin-based drivers) to easily add new connection types without modifying the system core.
- Ensure secure and auditable operations through service and admin access separation, as well as comprehensive logging.

Expected Outcomes

- A centralized control point (UI + API) for managing data connections and triggers.
- Operators can request and receive query results via WhatsApp without direct access to the database or historian systems.
- Easy addition of new connections or drivers through the plugin system, along with real-time connection status monitoring (health checks).
- Stable and reliable system operation, ensuring persistent WhatsApp sessions, automatic plugin reconnection, and complete activity logging.

2.2. Project context

The WA-Integration project was initiated to address operational needs for simplifying access to monitoring data. The operations team required an easier way to receive data and query results directly through WhatsApp without logging into historian or database systems. The main goal of the project is to provide a centralized platform that connects data sources such as AVEVA/PI, MySQL/Oracle, and AI services with a WhatsApp bot and a web interface for creating and executing triggers. Within the organization, this project functions as an operational solution or middleware that supports digitalization of operations and incident response, rather than as a general end-user product. Its modular architecture (plugin/AI-based) also enables further integration with other ongoing or future initiatives.

2.3. Previous activities

- Initialization & Scaffolding — Set up the project framework, including backend, frontend, and bot components. *(completed)*
- Architecture & Plugin Design — Developed a modular system to support the addition of new data connections. *(completed)*

- Data Connection & Auto-Connect — Implemented automatic server connection to data sources with health checks. *(completed)*
- Trigger Engine & Migration — Built the trigger rules system and migrated data storage to a database. *(in progress)*
- Backend API & Routing — Developed APIs to enable interaction between the frontend, bot, and backend system. *(in progress)*
- WhatsApp Bot Integration — Implemented the bot to handle user commands and send notifications. *(completed)*
- Frontend Management UI — Designed the web interface for managing data connections and triggers. *(completed)*
- AI Integration — Added AI plugins to enable intelligent features when required. *(completed)*
- Debugging & Patching — Conducted continuous bug fixing and iterative stability improvements. *(in progress)*
- Data Migration & Hardening — Migrated data to the main database, applied masking, and strengthened security. *(in progress)*

2.4. Timeline

The WA-Integration project began in late August and has been progressing gradually without a fixed deadline. The development follows a continuous improvement approach, focusing on stability and modularity. The following outlines the progress achieved so far:

- What Initialization & Scaffolding — Set up the project framework, including backend, frontend, and bot components. *(completed)*
- Architecture & Plugin Design — Developed a modular system to support the addition of new data connections. *(completed)*
- Data Connection & Auto-Connect — Implemented automatic server connection to data sources with health checks. *(completed)*
- Trigger Engine & Migration — Built the trigger rules system and migrated data storage to a database. *(in progress)*
- Backend API & Routing — Developed APIs to enable interaction between the frontend, bot, and backend system. *(in progress)*
- WhatsApp Bot Integration — Implemented the bot to handle user commands and send notifications. *(completed)*
- Frontend Management UI — Designed the web interface for managing data connections and triggers. *(completed)*
- AI Integration — Added AI plugins to enable intelligent features when required. *(completed)*
- Debugging & Patching — Conducted continuous bug fixing and iterative stability improvements. *(in progress)*
- Data Migration & Hardening — Migrated data to the main database, applied masking, and strengthened security. *(in progress)*

2.5. Budget

Please roughly outline the budget that your potential supplier could expect to work with to complete this project.

3. USERS

How would you describe the users of the new software? What is important for them, what are their needs and problems?

3.1. User groups / System's Actors

- Admin: Manages the entire system through the WA-Integration frontend, including adding or modifying data connections, creating or editing triggers, managing users and API keys, and viewing audit logs.
- User (Operator): Interacts only through the WhatsApp bot to request data, without access to the web-based admin interface.

.

3.2. Permissions

Model: A simple Role-Based Access Control (RBAC) system with two primary roles — Admin and User.
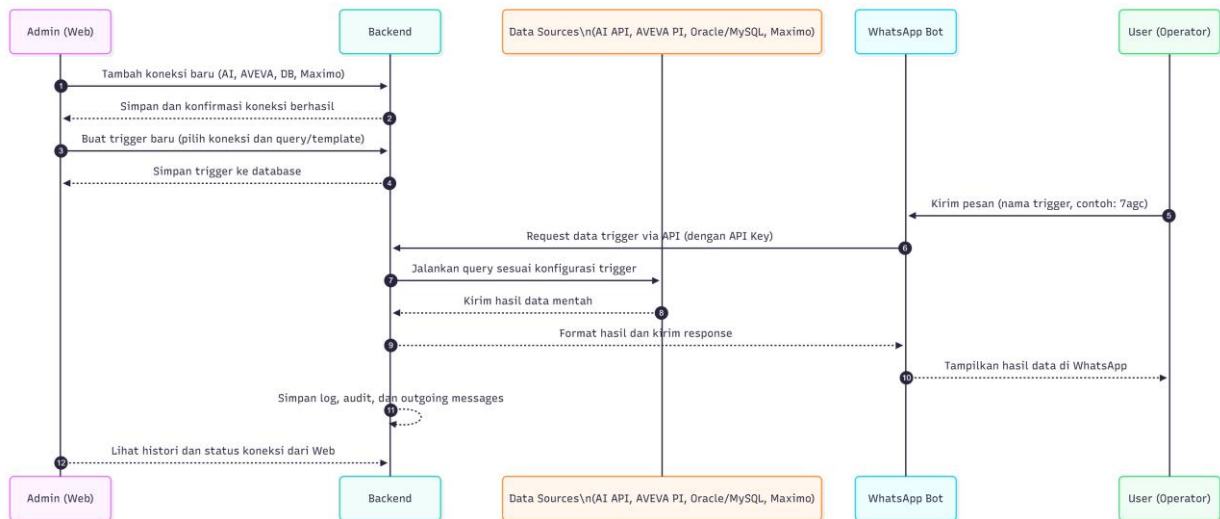
- Admin: Has full access to the WA-Integration frontend, including all management pages and APIs.
- User: Interacts only through the WhatsApp bot and does not have access to the web-based admin interface.

4. FUNCTIONAL REQUIREMENTS

This section provides information on all known software functions and outcomes of its use.
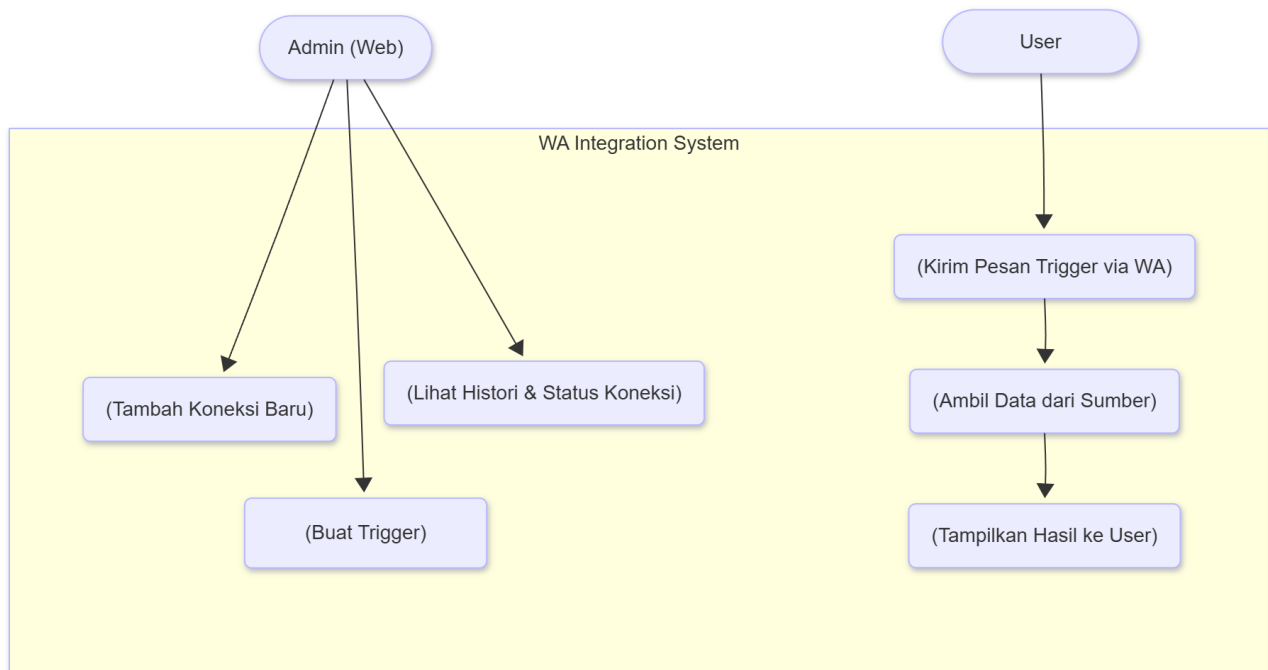
4.1. Business process

Please describe or present graphically steps to take to achieve the required business goal. For example:

4.2. Use cases – user functionalities

For every user group, please define or present graphically the actions they will perform in the software. For example:

4.3. Back office

Please describe additional functionalities which should be available for the administrator.

4.4. Reports

This point covers information about required data in the form of reports, statistics or other forms of data aggregation and presentation.

Please describe reports which are required from your software. In particular please consider what data should be included in each of them and whether data grouping or summing is required. Please also define the expected form of reports – only shown in the program or exportable and non-editable/ editable (e.g. as a pdf/ Excel file). If possible, please provide an example of each report.

4.5. Integrations

Please provide a list of programs with which you anticipate the need for integration. How is communication with these programs possible? Do you have documentation for these programs? If possible, please provide it with this document.

4.6. Notifications (mail, in app, text)

If you want users to receive notifications from your new software, please describe how and when.

4.7. Priorities

Please describe which of the above requirements are necessary to complete and which are optional or can be done at a later stage of the project.

4.8. Out of scope

If there are any elements of the software that do not fall within the scope of the project, please describe them here.

5. NON-FUNCTIONAL REQUIREMENTS

5.1. Limitations

Please describe any limitations that apply to your new software. Please list any legal standards and company's policies the software must be compliant with.

5.2. Security

Besides standards resulting from good software development practices, do you need to implement any specific security procedures?

Do you require your potential supplier to undergo an additional security audit?

5.3. Usability and accessibility

Do you have any specific guidelines or requirements when it comes to usability (e.g. resulting from legal norms)? Does the software have to meet the requirements of any of the Web Content Accessibility Guidelines (WCAG) levels?

5.4. Performance

Please describe what system load you anticipate - how many documents, users, transactions will be performed simultaneously and in a month. Please also define what maximum system response times are satisfactory for you, e.g. the home page loads in 2 seconds.

5.5. Localisation

What languages should your software be available in? Do you plan to extend this list in the future? Should the software take into account time differences and different formats for storing dates and numbers depending on the region in which it is to be used?

5.6. Training

Do you expect to run training in the operation of the new software? If so, for how many people and where would it take place? What form would the training be in (group/one-to-one/online training, or a written user's manual only)?

5.7. Design

Should the quote also include graphic design? If so, please attach your company's brand guidelines. If you already have a design – please attach it to the document. It is worth considering usability testing with a control group?

6. TECHNICAL REQUIREMENTS

6.1. Technologies

What are your requirements in terms of technologies to use? Do you need your potential supplier to suggest or recommend the most appropriate technologies?

6.1. Technological requirements

What equipment should the software support? Which operating system and what screen resolution? Is the software supposed to be running in an existing environment? Do you already have the infrastructure in place to implement the software? If not, should the potential supplier also provide a quote for the infrastructure? Please also describe infrastructure requirements.

6.2. Language versions

In what language versions will the software be running? Are there any changes planned in the future?

6.3. Data migration

Does the software require data migration? When and how will this data be made available?

6.4. Integration with external systems

Please provide a list of systems with which you anticipate the need for integration. How is communication with these systems possible?

7. IMPLEMENTATION REQUIREMENTS

7.1. Project management

Please describe your requirements and preferences for project management, methodology used, the means and frequency of contact.

7.2. Development process

Please describe your requirements for the development process. Should the software be available to you instantly and any time? How long does testing take on your side? Do you have your own procedures for the 'go live'?

7.3. Maintenance requirements

What support do you expect from your potential supplier when your new software is up and running? How long should it last for? Should the cost be included as part of this project or priced independently?

8. ADDITIONAL INFORMATION

Please add any other requests and information which you think might be relevant, useful or interesting for your potential supplier to know at this stage.