



Document	Software Project Plan	Release	17/08/26 V1.0 Final
Project	BRN WMS	Project code	200-WMS-26-001-00
Project period	05/01/26–24/08/26		

Revision history

Version	Date	Change	Prepared by
V1.0	17/08/26	Initial controlled issue. Consolidates the 05/01/26 planning baseline with lifecycle, quality, equipment-evidence, and configuration outcomes recorded through closure preparation.	Apirach Supattaratpateep

Sections 6, 8.5, 12.2 and 16 describe outcomes as at 17/08/26; the remaining sections retain the 05/01/26 planning baseline.

1. Purpose

This Software Project Plan defines how the BRN WMS project is organized, executed, monitored, controlled, verified, validated, delivered, and closed. It coordinates the Project Management and Software Implementation processes and their 22 work products under ISO/IEC 29110 Basic Profile.

2. Project overview

BRN WMS is a browser-based, multi-company and multi-warehouse management system. It centralizes warehouse master data, inventory movements, sales and purchasing documents, finance and accounting records, reports, access control, and operational notifications.

The project is intended to:

- Improve the accuracy and timeliness of warehouse operations.
- Provide current stock visibility across authorized warehouses and locations.
- Preserve lot, serial-number, expiry-date, and movement traceability.
- Control sales, purchasing, return, invoice, receipt, payment, and accounting workflows.
- Separate company data and restrict functions according to authorized roles.
- Provide management reports, operational alerts, and auditable records.

3. Scope

3.1 Included scope

- Company, user, role, application-access, SMTP, and system configuration.
- Contact, product, category, warehouse, storage-area, and bin master data.
- Stock-in, stock-out, stock transfer, balance, lot, serial number, and expiry control.
- Warehouse capacity, occupancy, movement, expired-stock, and product-lot reporting.
- SKU and warehouse-location barcode labels and supported scanning workflows.
- Quotations, sales orders, returns, invoices, and credit notes.
- Purchase requests, purchase orders, purchase invoices, and supplier returns.
- Receipt billing, receipts, payment billing, and payments.
- Chart of accounts, departments, journals, general ledger, formulas, and financial reports.
- Controlled document numbering and lifecycle/status handling.
- Real-time notifications using Node.js and Socket.IO.
- Scheduled stock/GL maintenance, low-stock alerts, and overdue-invoice alerts.
- Deployment configuration, database setup, operating guidance, and maintenance information.
- ISO/IEC 29110 work products and controlled project repository.

3.2 Excluded scope

- Procurement of servers, networks, barcode scanners, printers, or user devices.
- Legacy-data migration unless separately assessed and approved.
- Integration with external ERP, banking, shipping, tax, or third-party services unless approved through change control.
- Custom functionality outside the baselined requirements.
- Production hosting or third-party subscription fees unless separately authorized.

4. Objectives and success criteria

The project is successful when:

- All acceptance-critical requirements are implemented and traceable to verification evidence.
- Representative warehouse, sales, purchasing, finance, and accounting workflows pass validation.
- No unresolved critical defect blocks intended operation or compromises security or data integrity.
- Installation, configuration, user, operation, and maintenance documentation is available.
- Software and controlled work products are stored in the project repository.
- Verification, validation, and acceptance records receive the required review and authorization.

5. Lifecycle and schedule

Phase	Period	Main activities	Primary outputs
Initiation	05/01/26– 23/01/26	Establish need, stakeholders, objectives, scope, and authority.	Statement of Work, stakeholder records

Phase	Period	Main activities	Primary outputs
Planning	12/01/26–18/02/26	Collect requirements; define schedule, resources, risks, controls, and baselines.	Customer Requirements, Work Schedule, Software Project Plan
Implementation	19/02/26–29/05/26	Analyze, design, code, configure, integrate, review, and test the system.	SRS, Software Design, Components, Test Cases, Software
Verification and validation	30/05/26–07/08/26	Review work products and validate representative operational workflows.	Traceability, Test Report, Verification and Validation Results
Documentation and stabilization	01/06/26–24/08/26	Prepare guides, close corrections, configure deployment, and prepare demonstration data.	User Documentation, Operation Guide, Maintenance Documentation
Closure	10/08/26–24/08/26	Review repository, resolve open actions, obtain acceptance, and close the project.	Acceptance Report, List of Evidence, repository baseline

Detailed activities and evidence are maintained in the Work Schedule.

6. Software development lifecycle methodology

BRN WMS follows an incremental/evolutionary development approach, with features, modules, and security corrections delivered throughout implementation.

BRN WMS is more accurately described as incremental/evolutionary, within the same overall lifecycle stages used for planning and reporting purposes (Section 5):

Stage	How it was actually approached
Requirements	Captured once at a level (Customer Requirements), then implicitly refined as implementation proceeded — e.g., the Rack → Bin terminology change (CH-001) and multiple onboarding/security corrections show requirements being clarified during implementation, not frozen beforehand
Design	Not produced as an upfront, separate artifact; Software Design (work product 12) was derived from the as-built architecture, not authored before coding began
Implementation	Continuous, feature-by-feature, evidenced by 100 commits across the implementation period (19/02/26–29/05/26; 110 in the repository overall) with no clean phase boundary between "build" and "test/fix"
Verification	Interleaved throughout implementation (ongoing manual exercising and correction, per the Correction Register) rather than concentrated in a single verification phase; formal independent verification remains a separate, not-yet-executed activity (work product 21)
Stabilization/closure	A distinct late-stage effort (03/08/26 onward) closer to a traditional stabilization phase

7. Organization and responsibilities

Role	Assigned person	Responsibilities
Project Sponsor / Customer Representative / Authorized Approver	Seri Viriyasakultorn	Represent customer needs; authorize scope, resources, baseline changes, acceptance, and project closure.

Role	Assigned person	Responsibilities
Project Manager	Apirach Supattaratpateep	Plan and monitor work; assign responsibilities; manage risks, issues, communication, changes, and closure.
System Analyst	Noppong Chareunsook	Analyze customer requirements, specify system behavior, and maintain technical traceability.
Developer	Thanakorn Sathitwitayakul	Design, implement, configure, correct, and maintain source code, aligned with Git implementation evidence.
Tester / Reviewer	Parin Ngamkham	Prepare and execute tests, review work products, report defects, and confirm corrections.
Document Control	Yaowalak Bangchomphoo	Control identifiers, versions, approvals, distribution, repository content, and evidence.

One person may perform more than one operational role when independence is not mandatory. Approval authority must remain with the designated authorized approver or a formally delegated authority.

8. Resources and environment

8.1 Schedule summary

Phase	Calendar span	Approx. duration
Initiation	05/01/26–23/01/26	19 days
Planning	12/01/26–18/02/26	38 days
Implementation	19/02/26–29/05/26	100 days
Verification and validation	30/05/26–07/08/26	70 days
Documentation and stabilization	01/06/26–24/08/26	85 days
Closure	10/08/26–24/08/26	15 days

These are calendar spans rather than effort estimates.

8.2 Software and infrastructure

- PHP 8.0 or later.
- MySQL or MariaDB.
- Nginx or another compatible PHP web server.
- Node.js, npm, Socket.IO, and PM2 for real-time and scheduled services.
- Git repository with `main` as the controlled integration baseline.
- Current standards-based desktop and mobile browsers.
- Asia/Bangkok time zone across application and scheduled-job environments.

8.3 Logical application components

Component	Purpose
PHP web application	Main user interface and operational APIs
Identity/company database	Users, companies, access, SMTP, usage, and related configuration
WMS/accounting database	Warehouse, inventory, documents, finance, and accounting records
Node.js notification service	Browser notifications and application event relay
Node.js scheduler	Aggregate maintenance and scheduled operational alerts

8.4 Configuration constraints

- Local configuration and secrets must not be committed to Git.
- Production database credentials must use the minimum privileges required after installation.
- Application, database, upload, and service configuration must follow the controlled configuration guide.
- Public access to source-control metadata, secrets, and internal service endpoints must be restricted.

8.5 Computer and equipment resources

The example reference package's Section 7 lists specific equipment (notebook count, scanner, Git server, software licenses). No equipment inventory record exists as project evidence for BRN WMS — this is not fabricated here. What is evidenced instead:

Item	Evidence
Git hosting	Primary remote origin (188.166.228.62:nok/wms-app.git) and backup remote backup (GitHub) — see Project Repository, work product 9
Deployment target	Manual LAMP install or Docker Compose stack (php-apache, mariadb, node/pm2) — see Software Configuration, work product 8, and Product Operation Guide, work product 19
Developer workstation(s)	Not recorded — no inventory evidence exists
Office/licensed software (word processor, spreadsheet, etc.)	Not recorded — no inventory evidence exists

9. Deliverables and work products

9.1 Project Management work products

1. Statement of Work
2. Project Plan, including Work Schedule, Software Project Plan, and Customer Requirements
3. Progress Status Record
4. Correction Register
5. Acceptance Report
6. Change Report
7. Meeting Record
8. Software Configuration

9. Project Repository
10. Project Repository Backup

9.2 Software Implementation work products

1. Software Requirements Specification
2. Software Design
3. Traceability Record
4. Software Components
5. Test Cases and Test Procedures
6. Test Report
7. Software
8. Software User Documentation
9. Product Operation Guide
10. Maintenance Documentation
11. Verification Results
12. Validation Results

10. Monitoring and communication

Record or activity	Frequency or trigger	Owner	Audience
Work Schedule update	At least weekly during active work	Project Manager	Project team and sponsor
Progress Status Record	Weekly during active work	Project Manager	Project team and sponsor
Project meeting	At planned reviews or when decisions are required	Project Manager	Relevant stakeholders
Meeting Record	For each formal project/review meeting	Recorder	Attendees and affected stakeholders
Correction Register	When a defect, issue, or nonconformity is identified	Tester / Project Manager	Assigned owner and reviewer
Change Report	When a baseline change is requested	Project Manager	Sponsor, affected team, customer representative
Repository review	At each baseline and project closure	Document Control	Project Manager and reviewer

Progress status includes completed work, planned work, deviations, risks, issues, required decisions, corrective actions, and schedule impact.

11. Risk management

ID	Risk	Impact	Planned response / control	Owner
R-01	Pre-development planning records were prepared after the fact	Audit evidence may be weaker than the records suggest.	Mark assumptions clearly and obtain review and approval.	Project Manager
R-02	Unauthorized access or cross-company data exposure	Confidentiality and integrity failure.	Server-side role guards, company/warehouse scoping, session controls, and security review.	Developer / Reviewer
R-03	Incorrect inventory balance	Operational and financial records become unreliable.	Transactions, input validation, locking, approval flow, reconciliation, and movement tests.	Developer / Tester
R-04	Secret or configuration exposure	System compromise or service interruption.	Ignore local secrets, provide templates, restrict web access, and review deployment configuration.	Configuration Manager
R-05	Incomplete requirement or acceptance evidence	Delivery cannot be objectively demonstrated.	Maintain bidirectional traceability and obtain signatures before baselining.	Project Manager
R-06	Uncontrolled scope expansion	Schedule and quality degradation.	Require Change Report, impact analysis, authorization, and re-planning.	Project Manager / Sponsor
R-07	Inadequate backup or recovery	Loss of source, documents, or operational data.	Maintain repository backup and documented database/file backup and recovery procedures.	Configuration Manager
R-08	Third-party or environment incompatibility	Deployment or notification failure.	Document supported versions and verify the target environment before acceptance.	Developer / System Administrator

Risks are reviewed with progress status. New risks and changes to exposure or response are recorded by the Project Manager.

11.1 Contingency actions for non-completed tasks

Distinct from the risk table above (which addresses project-level threats), this table addresses the specific, recurring situation of an individual task or work product not finishing by its planned date.

Situation	Common cause	Likely impact	Contingency action	Owner
Task not finished by its due date	Timeline or resource estimate was too tight	Downstream tasks slip; delivery date at risk	Meet with the team to reassess status and priority; agree and communicate a revised timeline	Project Manager
Delay due to insufficient staffing (illness, unavailability)	Single-person roles (Section 7) have no backup	Work stalls until the person returns	Document a handover/knowledge-transfer note; consider temporary outside help for the specific gap only with Sponsor authorization	Project Manager

Situation	Common cause	Likely impact	Contingency action	Owner
Delay due to late or incomplete requirement/content input	Upstream dependency (customer input, data) not ready	Downstream development or testing cannot proceed	Escalate to the Sponsor with a clear description of what is blocking; agree a revised input date	Project Manager
Delay due to unexpected technical complexity	Underestimated integration/defect complexity discovered during work	Task takes materially longer than planned	Prioritize by severity (Critical/High/Low, per Section 12.1); temporarily set aside lower-priority items	Developer
Delay because scope changed mid-task	Requirement changed after work started	Effort already spent may be partially invalidated	Raise a Change Report; obtain authorization before continuing under the new scope	Project Manager

12. Quality assurance

- Assign a unique identifier to each approved requirement.
- Review requirements for clarity, completeness, consistency, testability, and scope alignment.
- Review design against requirements and operational constraints.
- Review implementation for authorization, tenant isolation, data integrity, configuration safety, and error handling.
- Trace requirements to design elements, components, test cases, and results.
- Record defects and nonconformities in the Correction Register.
- Re-test corrected behavior and retain objective evidence.
- Validate representative end-to-end scenarios with customer-oriented data.

12.1 Defect disposition

Severity	Meaning	Acceptance treatment
Critical	Prevents core operation, compromises security, causes cross-company exposure, or corrupts essential data.	Must be corrected and verified before acceptance.
Major	Material function fails without an acceptable workaround.	Correct before acceptance or obtain explicit authorized disposition.
Minor	Limited impact with an acceptable workaround.	Record planned correction or authorized acceptance.
Observation	Improvement or documentation item without functional failure.	Record and prioritize as appropriate.

12.2 Quality criteria and evaluation methods

The example reference package states measurable quality thresholds directly in the Software Project Plan (response time, uptime, security scan result, etc.). BRN WMS's measurable criteria are already defined as non-functional requirements in Customer Requirements (Section 8) and restated as technical requirements in the SRS (work product 11); this table cross-references them here rather than duplicating a second copy that could drift out of sync.

Quality area	Criterion	Evaluation method	Reference
Functional correctness	All Must requirements implemented and traceable	Traceability Record review + Test Report execution	NFR-009, work products 13, 16
Performance	Practical operational response time; aggregates support dashboards/reports	Representative-operation timing under agreed data volume	NFR-006, SR05:001–003, TC-NFR-006
Security	No unresolved critical security defect; server-side auth/tenant scope enforced	Negative-authorization/invalid-input testing	NFR-002, SR07:001–005, TC-NFR-002
Reliability/integrity	No invalid negative/duplicate stock or GL movement	Failure/rollback and concurrency test	NFR-003, SR09:003, TC-NFR-003
Usability	Responsive UI usable on desktop and warehouse-floor devices	Representative-screen check at agreed viewport sizes	NFR-005, SR02:002, TC-NFR-005
Installability	Installation/configuration/backup/recovery repeatable	Follow Product Operation Guide end-to-end	NFR-004, TC-NFR-004
Post-delivery support	Backup and restoration	Restoration check performed manually by the Developer	Project Repository (Backup), work product 10

Actual results belong in the Test Report and Verification/Validation Results, which show 34 of 34 passed and 12 of 12 passed, executed 10/08/26–14/08/26 (see the Acceptance Report's current-status note).

13. Verification and validation

Verification confirms that each work product satisfies its specified inputs and criteria. Validation confirms that the integrated BRN WMS supports its intended operational use.

Verification covers:

- Customer Requirements and Software Requirements Specification.
- Software Design and database/configuration design.
- Software Components and integration behavior.
- Test Cases, Test Procedures, Test Report, and traceability.
- User, operation, configuration, and maintenance documentation.

Validation covers representative workflows for:

- User onboarding and role-based access.
- Warehouse and product configuration.
- Stock receipt, issue, transfer, balance, lot, serial, and expiry handling.
- Sales, purchasing, return, invoice, receipt, and payment workflows.
- Accounting postings and management reports.
- Notifications, scheduled jobs, and operational recovery.

14. Configuration management

Configuration items include:

- PHP, JavaScript, CSS, Node.js, and database/setup source files.
- Application and service configuration templates.
- Database schema and migration/setup logic.
- Controlled requirements, design, test, guide, and management work products.
- Approved releases, evidence, and repository backups.

Controls include:

- Controlled main baseline.
- Project-code-based filenames and document version/status identifiers.
- Review and authorization before changing an approved baseline.
- Exclusion of passwords, tokens, local configuration, logs, uploads, and generated secrets from source control.
- Repository and operational-data backup with recoverability checks.

15. Change and correction control

A baseline change follows this sequence:

1. Record the requested change and reason.
2. Analyze its scope, schedule, technical, quality, security, and documentation impact.
3. Obtain authorization from the designated authority.
4. Update affected plans, requirements, design, traceability, tests, and configuration records.
5. Implement and verify the change.
6. Record the result and close the Change Report.

Defects and nonconformities are recorded in the Correction Register, assigned to an owner, corrected, re-tested, and closed with evidence.

16. Repository and document control

16.1 File naming convention

BRN WMS's actual naming convention, in force since PM work product 1 and used consistently across all 46 controlled files, differs deliberately from the example reference project's:

[Project code] [Document name] [YYYYMMDD Buddhist] V[version], for example 200-WMS-26-001-00 Correction Register 25690817 V1.0.

Element	Meaning	Example
Project code	200-WMS-26-001-00	Fixed
Document name	Descriptive title; where a work product has multiple instances (Progress Status Record, Change Report, Meeting Record), a short distinguishing suffix is added	- Rack to Bin Rename

Element	Meaning	Example
Date	Compact Buddhist-calendar date (YYYYMMDD), matching the date on the document header	25690817
Version	V<major>.<minor>, e.g. V1.0	V1.0

BRN WMS document filenames do not append author initials.

16.2 Version declaration

Documents created for BRN WMS are released directly at V1.0 once content is complete, without the example's separate 0.1/0.2 Draft stages — BRN WMS work products are not labeled Draft unless explicitly requested, per the same recorded decision. "Final" in the document-control Status field means the controlled content is complete; authorization and the project-acceptance decision are recorded in each document's Approval section and the Acceptance Report.

16.3 Repository and backup

- Every controlled filename starts with 200-WMS-26-001-00.
- Documents identify title, date, version, status, preparer, reviewer, and approver as applicable.
- Drafts remain distinguishable from approved baselines.
- The Project Repository contains the current controlled work products and supporting evidence (work product 9).
- The repository backup is maintained separately and checked during project closure (work product 10); the baseline and restoration checks were performed manually by the Developer (BK-001 and BK-002 closed), package completeness is recorded by BK-003, and final SDLC delivery-branch synchronization is tracked by BK-004 through 24/08/26.
- Superseded records are retained or archived according to organizational control practices.

17. Acceptance and closure

The project may close when:

- Acceptance-critical requirements have passed their linked verification and validation.
- No unresolved critical defect remains.
- Deferred items and accepted exceptions have authorized dispositions.
- Required user, operation, configuration, and maintenance documents are available.
- Software, source, setup information, evidence, and required work products are in the repository.
- The Acceptance Report and closure decision are signed by the authorized approver.

18. Approval

Prepared by

Name: Apirach Supattaratpateep

Role: Project Manager

Signature: _____

Date: _____

Technical contributor

Name: Thanakorn Sathitwitayakul

Role: Developer

Signature: _____

Date: _____

Reviewed and authorized by

Name: Seri Viriyasakultorn

Project roles: Project Sponsor / Customer Representative / Authorized Approver

Position: Managing Director

Company: B.R.N. Enterprise Co., Ltd.

Signature: _____

Date: _____