

K&N TECHUB

Farrago ERP

An Integrated ERP System for Bag & Garment Manufacturing

Software Requirements Specification

(IEEE Std 830-1998)

Prepared by: Kavishka Jayasinghe

Version: 1.0

Status: Draft

Date: June 17, 2026

Confidential — for internal use

Document Control

Document information

Title	Farrago ERP — Software Requirements Specification
Prepared by	Kavishka Jayasinghe
Organisation	K&N Techub
Version	1.0
Status	Draft
Date	June 17, 2026

Revision history

Version	Date	Author	Description
1.0	June 17, 2026	Kavishka Jayasinghe	Initial specification.

Executive Summary

Farrago ERP is a web-based Enterprise Resource Planning system built for a bag and garment manufacturing factory. It unifies the day-to-day operational functions of the factory — human resources, attendance, payroll, inventory, procurement, accounting, sales and customer management — into a single, role-secured platform, replacing the disconnected spreadsheets and manual ledgers typically used by small and medium manufacturers.

A defining characteristic of the system is *deep cross-module integration*: a single business action automatically propagates the records it implies across other modules. Recording a sales invoice draws down finished-goods stock and posts the revenue to the accounts; marking a pay period as paid posts the corresponding payroll expense; purchasing or restocking raw materials posts the procurement expense. This keeps the factory’s financial position continuously accurate without duplicate data entry.

This Software Requirements Specification documents the complete functional and non-functional requirements of the system in accordance with IEEE Std 830-1998. It describes the product perspective, the classes of users and their privileges, the requirements of each module, the external interfaces (Firebase Authentication, MongoDB Atlas, and the REST API), and the quality attributes the system must satisfy, together with the supporting use-case, data and architectural models.

Contents

Document Control	i
Executive Summary	ii
1 Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Definitions, Acronyms and Abbreviations	2
1.4 References	3
1.5 Overview of the Document	4
2 Overall Description	5
2.1 Product Perspective	5
2.2 Product Functions	5
2.3 User Classes and Characteristics	6
2.4 Operating Environment	7
2.5 Design and Implementation Constraints	7
2.6 Assumptions and Dependencies	8
3 System Features and Functional Requirements	9
3.1 Authentication and Access Control (AUTH)	9
3.1.1 Description and Priority	9
3.1.2 Stimulus / Response Sequences	9
3.1.3 Functional Requirements	9
3.2 Human Resources (HR)	10
3.2.1 Description and Priority	10
3.2.2 Stimulus / Response Sequences	11
3.2.3 Functional Requirements	11
3.2.4 Use Case: Add Employee	11
3.3 Attendance and Piece Capture (ATT)	12
3.3.1 Description and Priority	12
3.3.2 Stimulus / Response Sequences	13
3.3.3 Functional Requirements	13
3.4 Payroll (PAY)	13
3.4.1 Description and Priority	13

3.4.2	Calculation Model	14
3.4.3	Functional Requirements	14
3.5	Inventory (INV)	15
3.5.1	Description and Priority	15
3.5.2	Functional Requirements	15
3.6	Suppliers and Procurement (SUP)	16
3.6.1	Description and Priority	16
3.6.2	Functional Requirements	16
3.7	Accounting (ACC)	17
3.7.1	Description and Priority	17
3.7.2	Functional Requirements	17
3.8	Sales and Invoicing (SAL)	17
3.8.1	Description and Priority	17
3.8.2	Stimulus / Response Sequences	18
3.8.3	Functional Requirements	18
3.9	Customer Management (CUS)	18
3.9.1	Description and Priority	18
3.9.2	Functional Requirements	19
3.10	Dashboard and Reporting (DSH)	19
3.10.1	Description and Priority	19
3.10.2	Functional Requirements	19
3.11	Cross-Module Integration (INT)	20
3.11.1	Description and Priority	20
3.11.2	Integration Rules	20
3.11.3	Functional Requirements	20
3.12	Payroll Approval Workflow (APR)	21
3.12.1	Description and Priority	21
3.12.2	Stimulus / Response Sequences	21
3.12.3	Functional Requirements	21
3.13	User and Role Administration (USR)	22
3.13.1	Description and Priority	22
3.13.2	Functional Requirements	22
3.14	Activity Notifications (NTF)	22
3.14.1	Description and Priority	22
3.14.2	Functional Requirements	23
4	External Interface Requirements	31
4.1	User Interfaces	31
4.2	Hardware Interfaces	32

4.3	Software Interfaces	32
4.4	Communication Interfaces	32
4.4.1	REST API Endpoint Catalogue	33
5	Non-Functional Requirements	37
5.1	Performance Requirements	37
5.2	Security Requirements	37
5.3	Reliability and Availability	38
5.4	Usability	38
5.5	Maintainability and Portability	38
5.6	Scalability	39
6	System Models	40
6.1	Use-Case Model	40
6.2	Data Model (Entity-Relationship)	41
6.3	Sequence Model: Create Sale	41
6.4	Deployment Architecture	42
A	Data Dictionary	46
A.1	Employee	46
A.2	PieceTask	46
A.3	Attendance	47
A.4	PayrollAdjustment	47
A.5	Supplier	47
A.6	InventoryItem	48
A.7	Customer	48
A.8	Sale	49
A.9	Transaction	49
B	Glossary	51
C	Proposed Future Enhancements	52

List of Figures

2.1	System context of Farrago ERP.	6
3.1	The sign-in screen, the entry point to the system.	10
3.2	The employee register.	12
3.3	Daily / bulk attendance capture.	14
3.4	Payroll calculation for a pay period.	15
3.5	A generated payslip PDF.	24
3.6	Inventory listing with low-stock indicators.	25
3.7	Adding an item, including an optional purchase cost.	25
3.8	Supplier management.	26
3.9	Accounting — transactions and summary.	26
3.10	The Profit & Loss statement PDF.	27
3.11	The sales dashboard with analytics.	28
3.12	Creating a new sale / invoice.	29
3.13	The customer registry.	29
3.14	The consolidated dashboard.	30
4.1	The collapsible navigation rail (expanded).	36
6.1	Use-case diagram of Farrago ERP.	43
6.2	Entity-relationship diagram.	44
6.3	Sequence diagram — Create Sale.	45
6.4	Deployment architecture.	45

List of Tables

1.1	Definitions, acronyms and abbreviations.	2
1.1	Definitions, acronyms and abbreviations.	3
2.1	User classes and their characteristics.	6
2.1	User classes and their characteristics.	7
3.1	Functional requirements — Authentication and Access Control.	10
3.2	Functional requirements — Human Resources.	11
3.3	Use case — Add Employee.	11
3.3	Use case — Add Employee.	12
3.4	Functional requirements — Attendance.	13
3.5	Functional requirements — Payroll.	14
3.5	Functional requirements — Payroll.	15
3.6	Functional requirements — Inventory.	16
3.7	Functional requirements — Suppliers.	16
3.8	Functional requirements — Accounting.	17
3.9	Functional requirements — Sales.	18
3.10	Functional requirements — Customers.	19
3.11	Functional requirements — Dashboard.	19
3.12	Cross-module integration rules.	20
3.13	Functional requirements — Integration.	20
3.14	Functional requirements — Payroll Approval.	21
3.14	Functional requirements — Payroll Approval.	22
3.15	Functional requirements — User Administration.	22
3.16	Functional requirements — Notifications.	23
4.1	Principal screens.	31
4.2	Software interfaces.	32
4.3	REST API endpoint catalogue.	33
4.3	REST API endpoint catalogue.	34
4.3	REST API endpoint catalogue.	35
6.1	Principal use cases and their actors.	40
6.2	Data collections.	41
A.1	Employee fields.	46

A.2	PieceTask fields.	46
A.3	Attendance fields.	47
A.4	PayrollAdjustment fields.	47
A.5	Supplier fields.	47
A.5	Supplier fields.	48
A.6	InventoryItem fields.	48
A.7	Customer fields.	48
A.8	Sale fields.	49
A.9	Transaction fields.	49
A.9	Transaction fields.	50
B.1	Glossary of domain terms.	51

1 Introduction

This chapter introduces the Software Requirements Specification (SRS) for the Farrago ERP system. It states the purpose of the document, defines the scope of the product, lists the terminology and references used throughout, and outlines the structure of the remainder of the specification.

1.1 Purpose

The purpose of this document is to provide a complete and unambiguous specification of the requirements for **Farrago ERP**, a web-based enterprise resource planning system for a bag and garment manufacturing business. The document is intended to:

- establish a shared understanding of the system between the client (factory management), the development team, and the evaluators;
- serve as the authoritative baseline against which the design, implementation and acceptance testing of the system are verified;
- record both the functional behaviour of each module and the non-functional quality attributes the system must satisfy.

The intended audience of this SRS includes:

- **Factory management** — to confirm that the requirements reflect the real operational needs of the factory;
- **Developers** — as the basis for system and detailed design;
- **Testers / evaluators** — to derive test cases and acceptance criteria;
- **Maintainers** — to understand the system when extending or correcting it after delivery.

1.2 Scope

The product specified by this document is named **Farrago ERP**. It is a single-factory, multi-user web application accessed through a standard web browser. The system covers the following functional areas:

- **Authentication & Access Control** — secure sign-in and role-based authorization for five user roles.
- **Human Resources** — employee master records and lifecycle.
- **Attendance** — daily attendance and per-task piece capture for production workers.
- **Payroll** — calculation of piece-rate and fixed-basic wages, allowances, bonuses and deductions, and generation of payslips and a master payroll report.
- **Inventory** — tracking of raw materials, finished goods and defective items, including reorder alerts and stock movements.
- **Procurement & Suppliers** — supplier records and the purchasing / restocking of materials.
- **Accounting** — recording of income and expense transactions and the production of a profit & loss statement.
- **Sales & Invoicing** — creation of customer invoices, payment tracking and a sales analytics report.
- **Customer Management** — a reusable registry of clients.
- **Dashboard & Reporting** — consolidated key figures, charts and exportable PDF reports.

The central benefit of the system is the *automatic propagation of business events* between modules, removing duplicate manual entry and keeping inventory and accounting records continuously consistent with operational activity.

Out of scope. The following are explicitly *not* part of the current product: integration with external tax or government e-filing systems; online customer-facing storefronts or e-commerce; bank or payment-gateway integration; multi-factory / multi-branch consolidation; and a native mobile application. These are noted as possible future enhancements in Appendix C.

1.3 Definitions, Acronyms and Abbreviations

Table 1.1: Definitions, acronyms and abbreviations.

Term	Definition
ERP	Enterprise Resource Planning — integrated management of core business processes.

Table 1.1: Definitions, acronyms and abbreviations.

Term	Definition
SRS	Software Requirements Specification (this document).
RBAC	Role-Based Access Control — authorization based on a user's assigned role.
Piece rate	A wage scheme paying a worker a fixed amount per completed unit of a defined task.
Fixed basic	A wage scheme paying a fixed monthly salary, prorated by days worked.
Finished good	A completed, sellable manufactured product (e.g. a bag).
Raw material	An input consumed during manufacturing (fabric, thread, zips).
Reorder level	A stock threshold below which a raw material is flagged for replenishment.
Invoice	A sales document recording goods sold to a customer and amounts due.
Transaction	An accounting entry of type income or expense.
Net payable	An employee's wage after allowances, bonuses and deductions.
JWT / ID token	A signed token issued by Firebase that proves a user's identity to the API.
Custom claim	A key/value attribute (e.g. the user's role) embedded in a Firebase ID token.
API	Application Programming Interface — here, the system's REST endpoints.
CRUD	Create, Read, Update, Delete — the basic data operations.

1.4 References

- [1] IEEE, *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Std 830-1998.
- [2] ISO/IEC/IEEE, *Systems and software engineering — Life cycle processes — Requirements engineering*, ISO/IEC/IEEE 29148:2018.
- [3] Next.js Documentation, Vercel Inc. — <https://nextjs.org/docs>.
- [4] Express.js Documentation — <https://expressjs.com>.
- [5] MongoDB & Mongoose Documentation — <https://www.mongodb.com/docs>.

- [6] Firebase Authentication Documentation, Google — <https://firebase.google.com/docs/auth>.

1.5 Overview of the Document

The remainder of this SRS is organised as follows:

- **Chapter 2, Overall Description**, presents the product perspective, the major functions, the user classes, the operating environment, design constraints and assumptions.
- **Chapter 3, System Features and Functional Requirements**, specifies the detailed behaviour of every module, organised feature by feature with numbered functional requirements.
- **Chapter 4, External Interface Requirements**, describes the user, hardware, software and communication interfaces.
- **Chapter 5, Non-Functional Requirements**, specifies the performance, security, reliability, usability and other quality attributes.
- **Chapter 6, System Models**, contains the use-case, data, sequence and architectural models.
- **Appendices** provide the data dictionary, a glossary, and a list of proposed future enhancements.

2 Overall Description

This chapter gives the high-level context for the requirements stated in detail in Chapter 3. It describes how the product relates to its environment, the principal functions it provides, the classes of users it serves, the technical environment in which it operates, and the constraints and assumptions under which it is built.

2.1 Product Perspective

Farrago ERP is a new, self-contained product rather than a member of an existing product family. It replaces a collection of manual and spreadsheet-based processes with a single integrated system. The product follows a three-tier client/server architecture:

- **Presentation tier** — a single-page web application built with Next.js (React) that runs in the user’s browser and renders all screens, forms, charts and reports.
- **Application tier** — a stateless REST API built with Express.js that enforces authentication and authorization, applies business rules, and orchestrates the cross-module integrations.
- **Data tier** — a MongoDB document database that stores all persistent records (employees, attendance, payroll, inventory, suppliers, transactions, sales and customers).

The system depends on two external services: **Firebase Authentication** for user identity and role claims, and **MongoDB Atlas** as the managed database host. The relationship between these elements is shown in Figure 2.1 (see Chapter 6 for the full architecture and data models).

2.2 Product Functions

At a high level the system provides the following major functions. Each is specified in full in Chapter 3.

- Authenticate users and authorize their actions by role.
- Maintain employee master data and employment status.
- Capture daily attendance and per-task piece counts.
- Calculate payroll and produce payslips and a master payroll report.

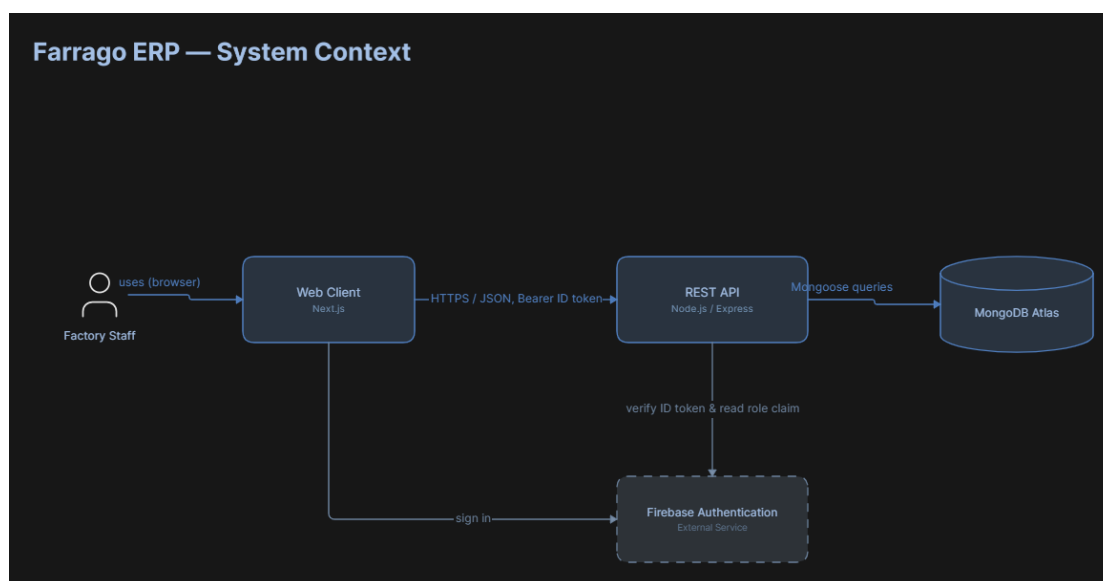


Figure 2.1: System context of Farrago ERP.

- Maintain inventory of raw materials, finished goods and defective items with low-stock alerts.
- Maintain supplier records and process purchases / restocks.
- Record income and expense transactions and produce a profit & loss statement.
- Create sales invoices, track payments, and produce a sales report.
- Maintain a reusable customer registry.
- Present a consolidated dashboard of key figures and trends.
- **Automatically propagate business events across modules** (sales, payroll and procurement each post the accounting and inventory records they imply).

2.3 User Classes and Characteristics

The system recognises five roles. A user's role is carried as a custom claim in their Firebase ID token and enforced by the API on every request. The **Administrator** role is a superuser that implicitly passes all role checks.

Table 2.1: User classes and their characteristics.

Role	Description	Principal privileges
Administrator (ADMIN)	Factory owner / system administrator. Highest expertise expected.	Full access to every module, including all sales, accounting and payroll functions.

Table 2.1: User classes and their characteristics.

Role	Description	Principal privileges
Manager (MANAGER)	Operations / finance manager.	HR, attendance, inventory, accounting, sales, payroll approval and reporting.
Supervisor (SUPERVISOR)	Production-floor supervisor.	HR and attendance entry, payroll calculation, inventory read/write; no sales or accounting write.
Maintainer (MAINTAINER)	Store / inventory keeper.	Attendance entry and inventory read; explicitly excluded from sales.
Worker (WORKER)	Production worker (subject of payroll).	Minimal / self-service access; not an operational data-entry role.

Users are assumed to be computer-literate at a basic level (comfortable with web forms and tables) but not technical specialists. The interface therefore favours clear labels, validation messages and confirmation prompts over dense or expert-oriented controls.

2.4 Operating Environment

- **Client.** Any modern desktop or laptop web browser (Chrome, Edge, Firefox) with JavaScript enabled. A network connection to the application server is required.
- **Application server.** A Node.js runtime hosting the Express REST API. The reference deployment runs on a managed cloud host (Render).
- **Database.** MongoDB (version 6 or later), hosted on MongoDB Atlas in the reference deployment.
- **Identity provider.** Firebase Authentication, accessed by the API through the Firebase Admin SDK for ID-token verification.
- **Frontend host.** The web client is served as a Next.js application (reference deployment on Netlify).

2.5 Design and Implementation Constraints

- **CON-1** The system shall be implemented as a web application using Next.js/React on the client and Express.js on the server.
- **CON-2** Persistent data shall be stored in MongoDB and accessed through the Mongoose object-document mapper.

- **CON-3** User authentication shall be delegated to Firebase Authentication; the API shall not store user passwords.
- **CON-4** Authorization shall be enforced server-side on every protected endpoint; client-side checks are advisory only.
- **CON-5** All monetary values shall be handled in Sri Lankan Rupees (LKR) and date/time bucketing for reports shall use the business timezone (Asia/Colombo).
- **CON-6** The REST API shall communicate using JSON over HTTPS.
- **CON-7** Generated documents (payslips, payroll, profit & loss and sales reports) shall be produced as PDF on the client.

2.6 Assumptions and Dependencies

- **AS-1** Each user has a Firebase account whose ID token carries a valid role custom claim.
- **AS-2** A reliable internet connection is available to clients and to the server's dependencies (Firebase, MongoDB Atlas).
- **AS-3** Piece-rate definitions and employee master data are configured before payroll is calculated.
- **AS-4** Finished-good inventory item names correspond to the descriptions used on sales invoices, so that stock can be drawn down automatically on a sale.
- **AS-5** The factory operates as a single business entity; multi-branch consolidation is not required.

3 System Features and Functional Requirements

This chapter specifies the functional requirements of the system, organised by module (system feature). For each feature it gives a description and priority, the principal stimulus/response sequences, and a numbered list of functional requirements. Functional requirements are identified as **FR-*MOD*-*n***, where *MOD* is a short module code and *n* a sequence number, so that they can be traced to design and test artefacts.

3.1 Authentication and Access Control (AUTH)

3.1.1 Description and Priority

The system must ensure that only registered users may access it and that each user may perform only the actions permitted to their role. Identity is established through Firebase Authentication; the API verifies the Firebase ID token on every request and reads the user's role from a custom claim. This feature is of **high priority** because every other feature depends on it.

3.1.2 Stimulus / Response Sequences

- **Sign in.** The user submits credentials; on success the client obtains a Firebase ID token and stores the session. Subsequent API calls carry the token in the **Authorization** header.
- **Protected request.** The client calls an endpoint; the API verifies the token, resolves the role, and either serves the request or rejects it with **401** (not authenticated) or **403** (insufficient role).
- **Sign out.** The user ends the session; the stored token is discarded and protected screens become inaccessible.

3.1.3 Functional Requirements

Table 3.1: Functional requirements — Authentication and Access Control.

ID	Requirement
FR-AUTH-1	The system shall allow a user to sign in using credentials managed by Firebase Authentication.
FR-AUTH-2	The system shall attach the user's Firebase ID token to every request made to the API.
FR-AUTH-3	The API shall reject any request to a protected endpoint that does not carry a valid, unexpired ID token, returning HTTP 401.
FR-AUTH-4	The API shall determine the user's role from the token's custom claim and reject requests whose role is not permitted for the endpoint, returning HTTP 403.
FR-AUTH-5	The system shall treat the ADMIN role as a superuser that satisfies every role check.
FR-AUTH-6	The system shall allow a signed-in user to sign out, after which protected resources are no longer accessible without signing in again.
FR-AUTH-7	The system shall never store user passwords in its own database.

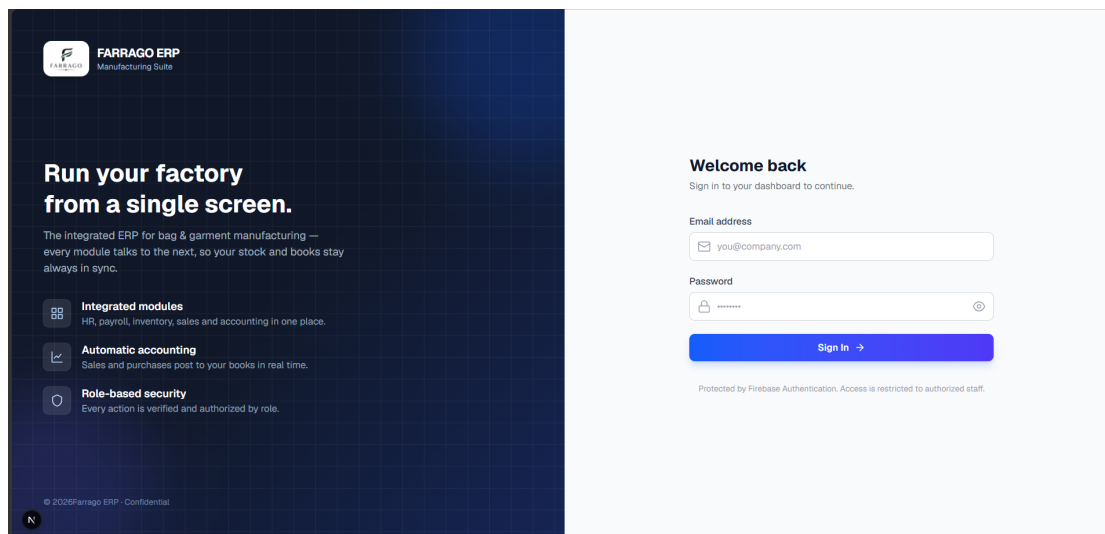


Figure 3.1: The sign-in screen, the entry point to the system.

3.2 Human Resources (HR)

3.2.1 Description and Priority

This feature maintains the master record of every employee — identification, role, department and wage scheme — which underpins attendance and payroll. It is of **high**

priority.

3.2.2 Stimulus / Response Sequences

- **Add employee.** An authorised user submits the employee's details and wage scheme; the system validates and stores a new record with a unique employee code.
- **Edit / deactivate employee.** An authorised user updates a record or marks an employee inactive so they are excluded from future payroll while their history is retained.
- **List / search employees.** A user views and filters the employee register.

3.2.3 Functional Requirements

Table 3.2: Functional requirements — Human Resources.

ID	Requirement
FR-HR-1	The system shall allow an authorised user to create an employee record capturing at least: name, employee code, role, department and wage scheme (piece-rate or fixed-basic).
FR-HR-2	The system shall enforce that each employee code is unique.
FR-HR-3	For a fixed-basic employee the system shall store a basic monthly salary.
FR-HR-4	The system shall allow an authorised user to update an existing employee record.
FR-HR-5	The system shall allow an employee to be marked inactive, excluding them from new payroll while preserving historical records.
FR-HR-6	The system shall allow users to list, search and filter employees by attributes such as department, role and status.
FR-HR-7	The system shall restrict create, update and deactivate operations to users holding an HR-authorised role (ADMIN, MANAGER, SUPERVISOR).

3.2.4 Use Case: Add Employee

Table 3.3: Use case — Add Employee.

Field	Detail
Use-case name	Add Employee
Actor	Administrator, Manager or Supervisor

Table 3.3: Use case — Add Employee.

Field	Detail
Pre-condition	The actor is signed in with an HR-authorized role.
Main flow	1. The actor opens the “Add Employee” form. 2. The actor enters the employee details and selects a wage scheme. 3. The system validates the input and that the employee code is unique. 4. The system stores the record and confirms success.
Alternative flow	If the employee code already exists or a required field is missing, the system rejects the submission and shows a validation message.
Post-condition	A new active employee record exists and is available to the attendance and payroll modules.

CODE	NAME	ROLE	PAY TYPE	BASIC	CONTACT	STATUS	ACTIONS
EMP-001	Chathura Ekanayake	HELPER	Piece	–	076 9409566	Active	Edit
EMP-002	Pradeep Herath	HELPER	Piece	–	073 7624747	Active	Edit
EMP-003	Nuwan Herath	HELPER	Piece	–	071 6491856	Active	Edit
EMP-004	Saman Ekanayake	MACHINE OPERATOR	Piece	–	078 5686284	Active	Edit
EMP-005	Pradeep Ekanayake	MACHINE OPERATOR	Piece	–	075 3524371	Active	Edit
EMP-006	Sunil Wickramasinghe	HELPER	Piece	–	074 5715235	Active	Edit
EMP-007	Ishara Senanayake	HELPER	Piece	–	073 1569732	Active	Edit
EMP-008	Nuwan Bandara	MACHINE OPERATOR	Piece	–	078 8150038	Active	Edit
EMP-009	Kamal Herath	MACHINE OPERATOR	Piece	–	072 3913067	Active	Edit
EMP-010	Ishara Perera	HELPER	Piece	–	070 2421415	Active	Edit

Figure 3.2: The employee register.

3.3 Attendance and Piece Capture (ATT)

3.3.1 Description and Priority

This feature records each employee’s daily attendance status and, for piece-rate workers, the quantity completed of each piece task on that day. Attendance is the source data for payroll. It is of **high priority**.

3.3.2 Stimulus / Response Sequences

- **Mark attendance.** An authorised user records, for a date, each employee's status (present, half-day, absent or leave).
- **Capture pieces.** For a piece-rate worker marked present, the user records the quantity completed of each piece task.
- **Bulk entry.** The user records attendance for many employees on a single date in one operation.

3.3.3 Functional Requirements

Table 3.4: Functional requirements — Attendance.

ID	Requirement
FR-ATT-1	The system shall allow an authorised user to record an attendance status of present, half-day, absent or leave for an employee on a given date.
FR-ATT-2	The system shall allow piece quantities to be captured per piece task against a present piece-rate worker's attendance record.
FR-ATT-3	The system shall support bulk attendance entry for multiple employees on a single date.
FR-ATT-4	The system shall prevent duplicate attendance records for the same employee on the same date.
FR-ATT-5	The system shall carry a per-record payment status (pending, paid or hold) used by the payroll lock workflow.
FR-ATT-6	The system shall restrict attendance entry to authorised roles (ADMIN, MANAGER, SUPERVISOR, MAINTAINER).

3.4 Payroll (PAY)

3.4.1 Description and Priority

This feature calculates wages for a chosen pay period from attendance and piece data, applies allowances, bonuses and deductions, and produces payslips and a master payroll report. It also locks a period as paid, which posts the payroll expense to accounting. It is of **high priority**.

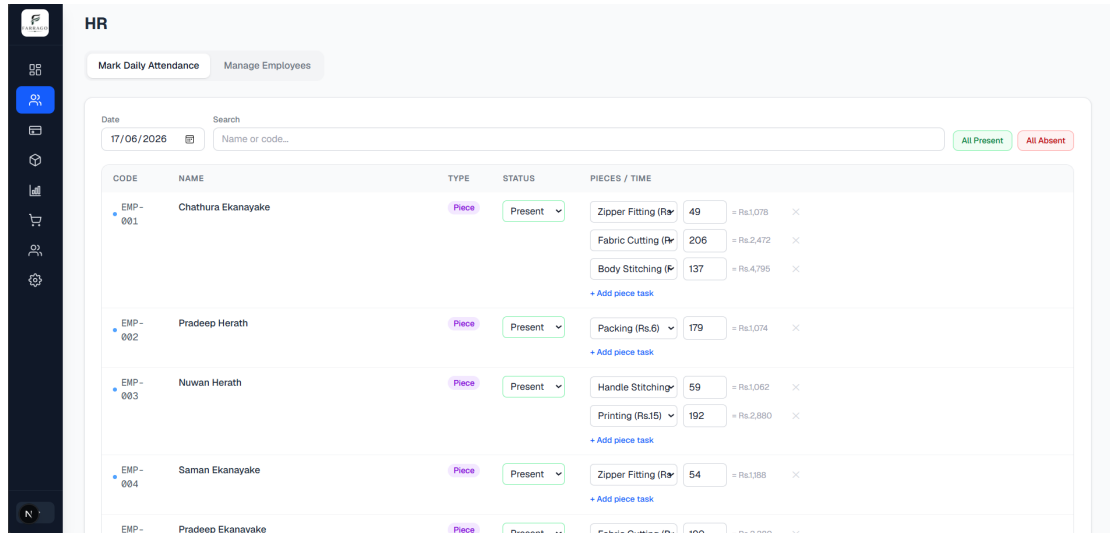


Figure 3.3: Daily / bulk attendance capture.

3.4.2 Calculation Model

Net payable for an employee over a period is computed as:

$$\text{Net} = \text{Gross} + \text{Allowances} + \text{Bonuses} - \text{Deductions}$$

where, for a **piece-rate** worker, $\text{Gross} = \sum_{\text{tasks}} (\text{quantity} \times \text{rate per piece})$, and for a **fixed-basic** worker, $\text{Gross} = \text{basic salary} \times \frac{\text{effective days}}{\text{calendar days}}$, with a half-day counted as 0.5 of an effective day. Allowances, bonuses and deductions are taken from monthly adjustment entries.

3.4.3 Functional Requirements

Table 3.5: Functional requirements — Payroll.

ID	Requirement
FR-PAY-1	The system shall calculate per-employee payroll for a date range from attendance, piece quantities and live piece-task rates.
FR-PAY-2	The system shall compute gross pay by wage scheme (piece-rate or prorated fixed-basic) as defined in the calculation model.
FR-PAY-3	The system shall apply allowance, bonus and deduction adjustments for the period to derive net payable.
FR-PAY-4	The system shall allow authorised users to create and delete monthly payroll adjustments (allowance, bonus, deduction).
FR-PAY-5	The system shall present period totals (employee count, total gross, total net payable) and a per-employee status of pending, paid or hold.

Table 3.5: Functional requirements — Payroll.

ID	Requirement
FR-PAY-6	The system shall allow an authorised user to mark the selected employees' period as paid (a strict pending/hold → paid transition).
FR-PAY-7	On marking a period paid, the system shall post a single accounting expense equal to the net payable of the newly-settled employees, and shall not post a duplicate when no employee actually transitions.
FR-PAY-8	The system shall generate a per-employee payslip PDF and a factory-wide master payroll report PDF.
FR-PAY-9	Payroll approval (mark-as-paid) shall be restricted to ADMIN and MANAGER.

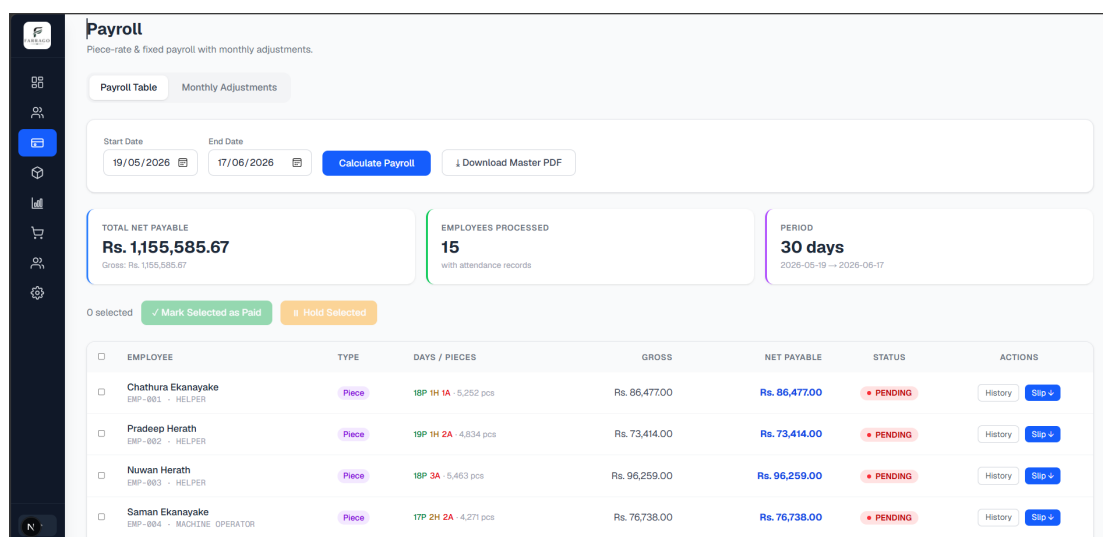


Figure 3.4: Payroll calculation for a pay period.

3.5 Inventory (INV)

3.5.1 Description and Priority

This feature tracks stock items in three classes — raw materials, finished goods and defective items — with quantities, units, reorder levels and an optional supplier link, and surfaces low-stock alerts. It is of **high priority**.

3.5.2 Functional Requirements

Table 3.6: Functional requirements — Inventory.

ID	Requirement
FR-INV-1	The system shall allow authorised users to create, update and delete inventory items classified as raw material, finished good or defective.
FR-INV-2	The system shall store for each item a quantity, unit, optional category, optional reorder level and optional supplier.
FR-INV-3	The system shall flag an item as low-stock when its quantity is at or below its reorder level (where a reorder level is set).
FR-INV-4	The system shall allow filtering of items by class, category and supplier, and a low-stock-only view.
FR-INV-5	The system shall provide a restock operation that atomically increases an item's quantity.
FR-INV-6	The system shall prevent stock quantities from becoming negative.
FR-INV-7	When a purchase or restock carries a total cost, the system shall post a corresponding accounting expense (see Section 3.11).
FR-INV-8	Inventory write operations shall be restricted to ADMIN , MANAGER and SUPERVISOR .

3.6 Suppliers and Procurement (SUP)

3.6.1 Description and Priority

This feature maintains supplier records and links them to the raw materials they provide. It is of **medium priority**.

3.6.2 Functional Requirements

Table 3.7: Functional requirements — Suppliers.

ID	Requirement
FR-SUP-1	The system shall allow authorised users to create, update and delete supplier records (name, company, contact number, address).
FR-SUP-2	The system shall allow an inventory item to reference a supplier.
FR-SUP-3	The system shall prevent deletion of a supplier that is still referenced by one or more inventory items.
FR-SUP-4	The system shall allow listing and filtering of suppliers.

3.7 Accounting (ACC)

3.7.1 Description and Priority

This feature records financial transactions as income or expense entries, categorises them, and produces a profit & loss statement over a chosen period. It is the financial backbone into which the other modules post. It is of **high priority**.

3.7.2 Functional Requirements

Table 3.8: Functional requirements — Accounting.

ID	Requirement
FR-ACC-1	The system shall allow authorised users to record an income or expense transaction with a date, category, amount, payment method and notes.
FR-ACC-2	The system shall record the source of each transaction (manual, sale, payroll or purchase) to distinguish automatically-posted entries from manual ones, without affecting manual entries.
FR-ACC-3	The system shall list and filter transactions by date range, type and category.
FR-ACC-4	The system shall allow deletion of a transaction.
FR-ACC-5	The system shall compute a period summary of total income, total expense and net profit, broken down by category.
FR-ACC-6	The system shall generate a profit & loss statement PDF for a selected period.
FR-ACC-7	Accounting functions shall be restricted to ADMIN and MANAGER .

3.8 Sales and Invoicing (SAL)

3.8.1 Description and Priority

This feature creates customer invoices with line items, computes totals, tracks payments and payment status, and provides sales analytics and a sales report. Creating an invoice draws down finished-goods stock and posts revenue to accounting. It is of **high priority**.

3.8.2 Stimulus / Response Sequences

- **Create invoice.** The user selects a customer (or enters a walk-in), adds finished-good line items with quantities and prices, applies discount/tax and any amount paid; the system computes totals, stores the invoice, decrements stock and posts income.
- **Record payment / settle.** The user records a further payment or marks the invoice paid in full; the system updates the balance and payment status (unpaid, partial, paid).

3.8.3 Functional Requirements

Table 3.9: Functional requirements — Sales.

ID	Requirement
FR-SAL-1	The system shall allow authorised users to create a sales invoice with a unique invoice number, a customer, and one or more line items.
FR-SAL-2	The system shall compute line totals, subtotal, discount, tax and grand total server-side; client totals shall not be trusted.
FR-SAL-3	The system shall track amount paid, balance due and a payment status of unpaid, partial or paid.
FR-SAL-4	The system shall allow recording additional payments and marking an invoice paid in full.
FR-SAL-5	On invoice creation, the system shall decrement matching finished-good stock and post the grand total as accounting income (see Section 3.11).
FR-SAL-6	The system shall provide sales analytics: revenue/collected/ pending totals, a daily trend, top customers, top products and monthly revenue.
FR-SAL-7	The system shall generate a sales report PDF.
FR-SAL-8	Sales functions shall be restricted to ADMIN and MANAGER; MAINTAINER is explicitly excluded.

3.9 Customer Management (CUS)

3.9.1 Description and Priority

This feature maintains a reusable registry of customers that can be selected when creating invoices. It is of **medium priority**.

3.9.2 Functional Requirements

Table 3.10: Functional requirements — Customers.

ID	Requirement
FR-CUS-1	The system shall allow authorised users to create, update and delete customer records (name, company, phone, email, address, notes).
FR-CUS-2	The system shall allow listing and searching of customers by name, company or phone.
FR-CUS-3	The system shall allow a registered customer to be selected when creating an invoice, pre-filling the customer details.
FR-CUS-4	Customer functions shall be restricted to ADMIN and MANAGER .

3.10 Dashboard and Reporting (DSH)

3.10.1 Description and Priority

This feature presents a consolidated overview of the factory — key figures, alerts and a financial trend — as the landing screen, and is the access point for the exportable PDF reports. It is of **medium priority**.

3.10.2 Functional Requirements

Table 3.11: Functional requirements — Dashboard.

ID	Requirement
FR-DSH-1	The system shall present headline figures: today's attendance, the month's net profit/loss, finished-goods count and low-stock alert count.
FR-DSH-2	The system shall present a daily income-versus-expense trend for the recent period, bucketed in the business timezone.
FR-DSH-3	The system shall list recent transactions and low-stock items.
FR-DSH-4	The dashboard shall be available to operational roles (ADMIN , MANAGER , SUPERVISOR).

3.11 Cross-Module Integration (INT)

3.11.1 Description and Priority

This is the defining capability of the system: a single business action automatically creates the records it implies in other modules, so inventory and accounting stay continuously consistent with operations without duplicate entry. It is of **high priority**.

3.11.2 Integration Rules

Table 3.12: Cross-module integration rules.

Trigger	Inventory effect	Accounting effect
Sales invoice created	Finished-good stock decremented per line item (floored at zero)	Income transaction posted for the grand total (category “Sales”)
Payroll period marked paid	—	Single expense posted for the net payable of the settled employees (category “Payroll”)
Raw material purchased / restocked with a cost	Quantity increased	Expense posted for the total cost (category “Raw Materials”)

3.11.3 Functional Requirements

Table 3.13: Functional requirements — Integration.

ID	Requirement
FR-INT-1	The integration side-effects shall be best-effort: a failure to post a derived record shall not roll back or block the primary action, and shall be surfaced as a non-fatal warning.
FR-INT-2	Each automatically-posted transaction shall be tagged with its source and a reference to the originating record.
FR-INT-3	The payroll posting shall be idempotent with respect to repeated mark-as-paid actions on an already-settled period.
FR-INT-4	Finished-good stock shall never be driven negative by a sale; shortfalls shall be reported as warnings.

3.12 Payroll Approval Workflow (APR)

3.12.1 Description and Priority

This feature introduces a two-step control over payment of wages. A Manager or Supervisor submits a selected set of employees and a pay period for payment approval; only an Admin may approve it, and approval is what actually settles the period and posts the payroll expense. It enforces separation of duties between preparing and authorising payment. It is of **high priority**.

3.12.2 Stimulus / Response Sequences

- **Submit.** A Manager/Supervisor selects employees for a period on the payroll screen and submits them for approval; the system snapshots the net payable and records a pending request without settling anything.
- **Approve.** An Admin opens the approvals queue and approves a request; the system settles the period (marks attendance paid, posts the expense) and marks the request approved.
- **Reject.** An Admin rejects a request; nothing is settled.

3.12.3 Functional Requirements

Table 3.14: Functional requirements — Payroll Approval.

ID	Requirement
FR-APR-1	The system shall allow a Manager or Supervisor to submit a set of employees and a pay period for payment approval, capturing a net-payable snapshot at submission time.
FR-APR-2	Submitting a request shall not settle attendance or post any accounting entry.
FR-APR-3	The system shall let an Admin view pending requests and either approve or reject each.
FR-APR-4	Approving a request shall settle the period — marking the relevant attendance PAID and posting the payroll expense per the payroll settlement model (Section 3.4).
FR-APR-5	Submitting shall be permitted to Admin, Manager and Supervisor; approving and rejecting shall be restricted to Admin.
FR-APR-6	A non-admin requester shall see only their own requests; an Admin shall see all requests.

Table 3.14: Functional requirements — Payroll Approval.

ID	Requirement
FR-APR-7	A request that is already approved or rejected shall not be decided again.

3.13 User and Role Administration (USR)

3.13.1 Description and Priority

This feature lets an Admin create and manage the user accounts and the role assigned to each. A user's role drives both which menu items they see and which API operations they may perform. It is of **high priority**.

3.13.2 Functional Requirements

Table 3.15: Functional requirements — User Administration.

ID	Requirement
FR-USR-1	The system shall let an Admin list all user accounts together with their assigned role.
FR-USR-2	The system shall let an Admin create a user account (email, password, optional display name) and assign one of the defined roles.
FR-USR-3	The system shall let an Admin change a user's role.
FR-USR-4	The system shall let an Admin delete a user account.
FR-USR-5	The system shall prevent an Admin from removing their own Admin role or deleting their own account.
FR-USR-6	A user's role shall determine which modules and menu items are visible to them and which API operations they are authorized to perform.
FR-USR-7	User administration shall be restricted to the Admin role.

3.14 Activity Notifications (NTF)

3.14.1 Description and Priority

This feature gives the Admin oversight of what other staff are doing: meaningful actions taken by non-admin users are logged and surfaced to the Admin as a notification feed

with an unread count. It is of **medium priority**.

3.14.2 Functional Requirements

Table 3.16: Functional requirements — Notifications.

ID	Requirement
FR-NTF-1	The system shall record an activity entry when a non-admin user performs a key action — creating a sale, submitting a payroll approval, logging attendance, adding an employee, creating or restocking inventory, or recording a transaction.
FR-NTF-2	Activity logging shall be best-effort and shall never block or fail the originating action.
FR-NTF-3	The system shall present the Admin a notification feed of recent activity together with an unread count.
FR-NTF-4	The Admin shall be able to mark notifications as read.
FR-NTF-5	The notification feed shall be restricted to the Admin role.


FARRAGO
 Garment Manufacturing

PAYSLIP
 Generated 6/17/2026

EMPLOYEE

Name Chathura Ekanayake

Code EMP-001

Role HELPER

Department FINISHING

Payment Type PIECE RATE

PAY PERIOD

From May 19, 2026

To June 17, 2026

Calendar Days 30

Days Present 18

Effective Days 18.5

Earnings

Piece Task	Quantity	Rate	Amount
Body Stitching	790 pcs	Rs. 35.00	Rs. 27,650.00
Printing	1,279 pcs	Rs. 15.00	Rs. 19,185.00
Zipper Fitting	648 pcs	Rs. 22.00	Rs. 14,256.00
Fabric Cutting	932 pcs	Rs. 12.00	Rs. 11,184.00
Handle Stitching	307 pcs	Rs. 18.00	Rs. 5,526.00
Packing	846 pcs	Rs. 6.00	Rs. 5,076.00
Quality Check	450 pcs	Rs. 8.00	Rs. 3,600.00
Gross Piece Earnings			Rs. 86,477.00

Allowances, Bonuses & Deductions

Gross Earnings	Rs. 86,477.00
Allowances (+)	Rs. 0.00
Bonuses (+)	Rs. 0.00
Deductions (-)	- Rs. 0.00

NET PAYABLE
Rs. 86,477.00

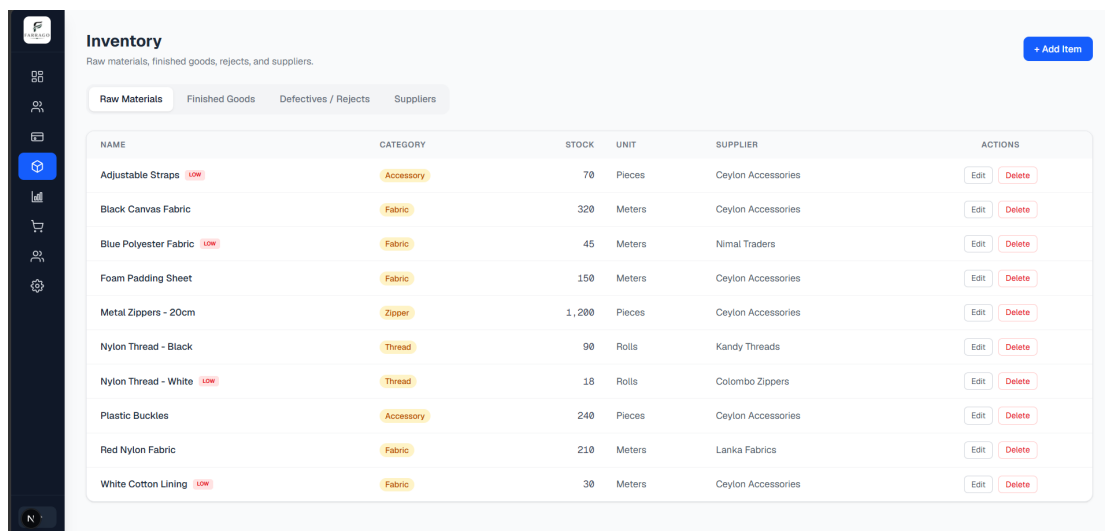
Present	Half Day	Absent	Leave	Effective
18	1	1	4	18.5

Prepared by _____

Employee Signature _____

System-generated salary slip · Farrago ERP

Figure 3.5: A generated payslip PDF.

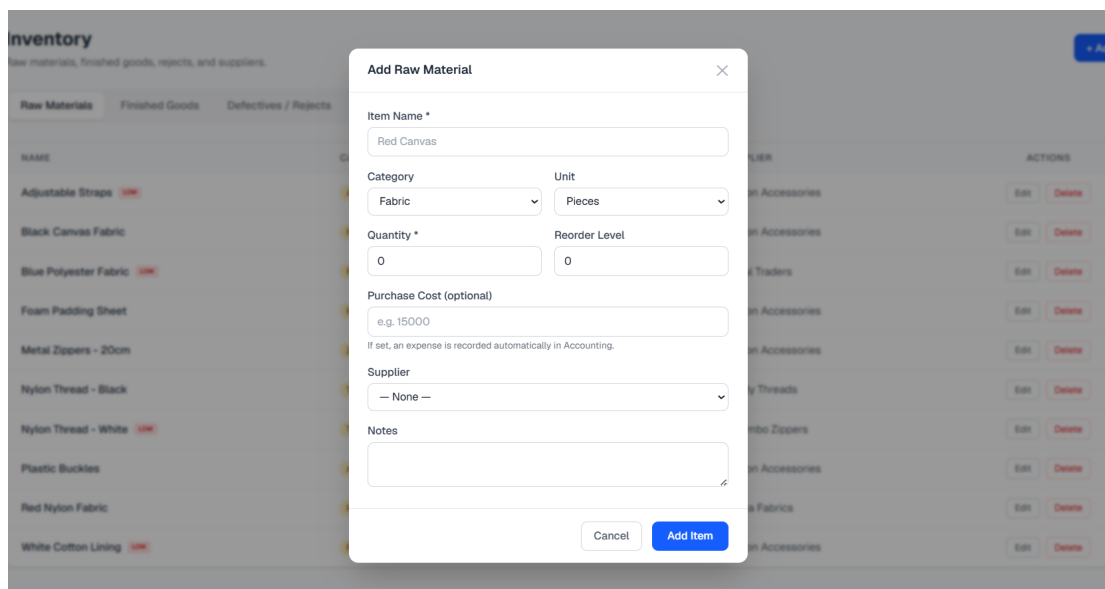


Inventory
Raw materials, finished goods, rejects, and suppliers.

Raw Materials Finished Goods Defectives / Rejects Suppliers

NAME	CATEGORY	STOCK	UNIT	SUPPLIER	ACTIONS
Adjustable Straps LOW	Accessory	70	Pieces	Ceylon Accessories	Edit Delete
Black Canvas Fabric	Fabric	320	Meters	Ceylon Accessories	Edit Delete
Blue Polyester Fabric LOW	Fabric	45	Meters	Nimal Traders	Edit Delete
Foam Padding Sheet	Fabric	150	Meters	Ceylon Accessories	Edit Delete
Metal Zippers - 20cm	Zipper	1,200	Pieces	Ceylon Accessories	Edit Delete
Nylon Thread - Black	Thread	90	Rolls	Kandy Threads	Edit Delete
Nylon Thread - White LOW	Thread	18	Rolls	Colombo Zippers	Edit Delete
Plastic Buckles	Accessory	240	Pieces	Ceylon Accessories	Edit Delete
Red Nylon Fabric	Fabric	210	Meters	Lanka Fabrics	Edit Delete
White Cotton Lining LOW	Fabric	30	Meters	Ceylon Accessories	Edit Delete

Figure 3.6: Inventory listing with low-stock indicators.



inventory
Raw materials, finished goods, rejects, and suppliers.

Raw Materials Finished Goods Defectives / Rejects

Add Raw Material

Item Name *
Red Canvas

Category: Fabric Unit: Pieces

Quantity *
0

Reorder Level
0

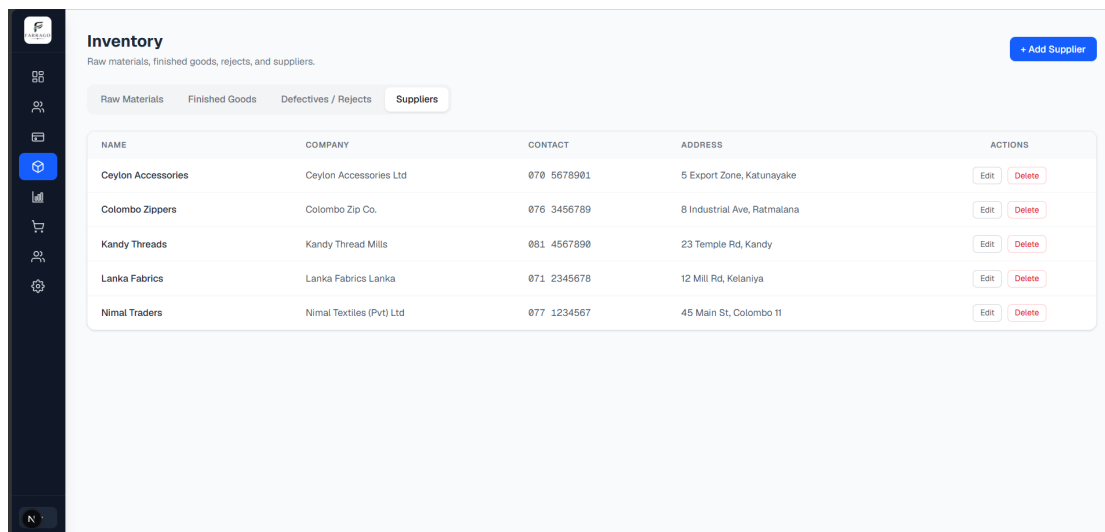
Purchase Cost (optional)
e.g. 15000
If set, an expense is recorded automatically in Accounting.

Supplier
— None —

Notes

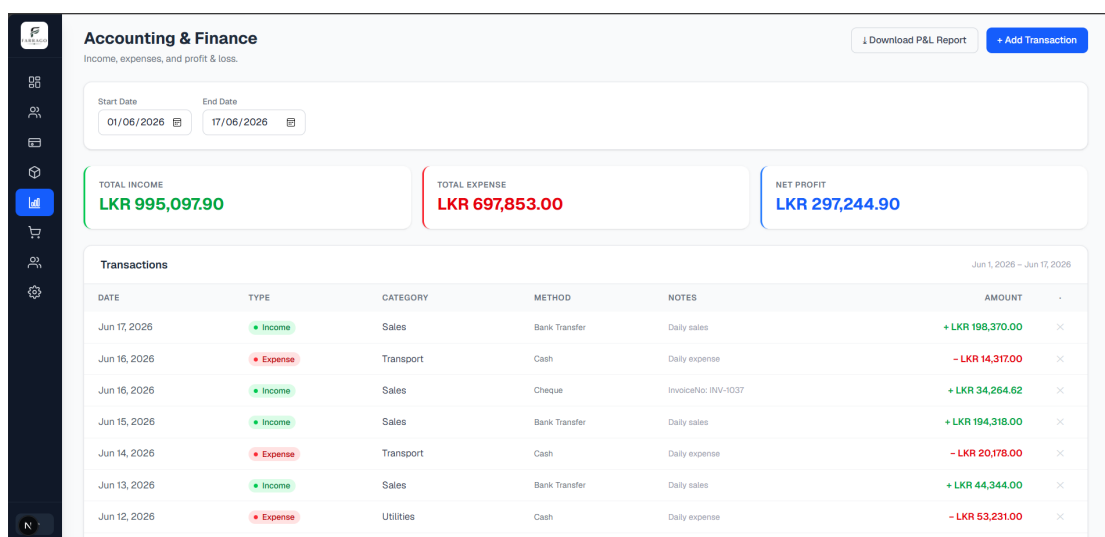
[Cancel](#) [Add Item](#)

Figure 3.7: Adding an item, including an optional purchase cost.



NAME	COMPANY	CONTACT	ADDRESS	ACTIONS
Ceylon Accessories	Ceylon Accessories Ltd	070 5678901	5 Export Zone, Katunayake	Edit Delete
Colombo Zippers	Colombo Zip Co.	076 3456789	8 Industrial Ave, Ratmalana	Edit Delete
Kandy Threads	Kandy Thread Mills	081 4567890	23 Temple Rd, Kandy	Edit Delete
Lanka Fabrics	Lanka Fabrics Lanka	071 2345678	12 Mill Rd, Kelaniya	Edit Delete
Nimal Traders	Nimal Textiles (Pvt) Ltd	077 1234567	45 Main St, Colombo 11	Edit Delete

Figure 3.8: Supplier management.



DATE	TYPE	CATEGORY	METHOD	NOTES	AMOUNT
Jun 17, 2026	Income	Sales	Bank Transfer	Daily sales	+ LKR 198,370.00
Jun 16, 2026	Expense	Transport	Cash	Daily expense	- LKR 14,317.00
Jun 16, 2026	Income	Sales	Cheque	InvoiceNo: INV-1037	+ LKR 34,264.62
Jun 15, 2026	Income	Sales	Bank Transfer	Daily sales	+ LKR 194,318.00
Jun 14, 2026	Expense	Transport	Cash	Daily expense	- LKR 20,178.00
Jun 13, 2026	Income	Sales	Bank Transfer	Daily sales	+ LKR 44,344.00
Jun 12, 2026	Expense	Utilities	Cash	Daily expense	- LKR 53,231.00

Figure 3.9: Accounting — transactions and summary.



FARRAGO

Garment Manufacturing

PROFIT & LOSS STATEMENT

June 1, 2026 to June 17, 2026

Generated 6/17/2026

Income

Category	Entries	Amount (LKR)
Sales	11	995,097.90
Total Income		995,097.90

Expenses

Category	Entries	Amount (LKR)
Payroll	1	252,436.00
Utilities	5	156,124.00
Transport	4	101,864.00
Rent	1	80,000.00
Raw Materials	2	77,973.00
Maintenance	1	29,456.00
Total Expenses		697,853.00

NET PROFIT

LKR 297,244.90

Total Income

LKR 995,097.90

Total Expenses

LKR 697,853.00

Net Profit

LKR 297,244.90

Chief Accountant

Farrago (Pvt) Ltd

System-generated Profit & Loss Statement - Farrago ERP

Figure 3.10: The Profit & Loss statement PDF.

[illegible]

New Sale / Invoice

Invoice No * Sale Date

Customer

Customer Name * Phone

Items *

+ Add line

Discount (LKR) Tax (%) Amount Paid

Subtotal **Rs. 0.00**
Grand Total **Rs. 0.00**
Balance Due **Rs. 0.00**

Figure 3.12: Creating a new sale / invoice.

Customers

Your client registry — reuse these on sales invoices.

NAME	COMPANY	PHONE	EMAIL	ACTIONS
City Fashion	City Fashion (Pvt) Ltd	073 9510084	cityfashion@example.lk	Edit Delete
Colombo Retail	Colombo Retail Group	074 6990192	colomboretail@example.lk	Edit Delete
Express Bags	Express Bags Lanka	071 9531112	expressbags@example.lk	Edit Delete
Galle Wholesale	Galle Wholesale Mart	078 2635810	gallewholesale@example.lk	Edit Delete
Jaffna Traders	Jaffna Trading Co.	073 2530586	jaffnatraders@example.lk	Edit Delete
Kandy Stores	Kandy Stores	076 9040041	kandystores@example.lk	Edit Delete
Kurunegala Mart	Kurunegala Mart	076 3969637	kurunegalamart@example.lk	Edit Delete
Lanka Bags Ltd	Lanka Bags Ltd	072 3005982	lankabagsltd@example.lk	Edit Delete
Matara Bag House	Matara Bag House	076 3497853	matarabaghouse@example.lk	Edit Delete
Negombo Distributors	Negombo Distributors	074 5074216	negombodistributors@example.lk	Edit Delete
Perera Traders	Perera & Sons	074 3250084	pereratradrs@example.lk	Edit Delete
Silva Stores	Silva Retail (Pvt) Ltd	074 6530823	silvastores@example.lk	Edit Delete

Figure 3.13: The customer registry.

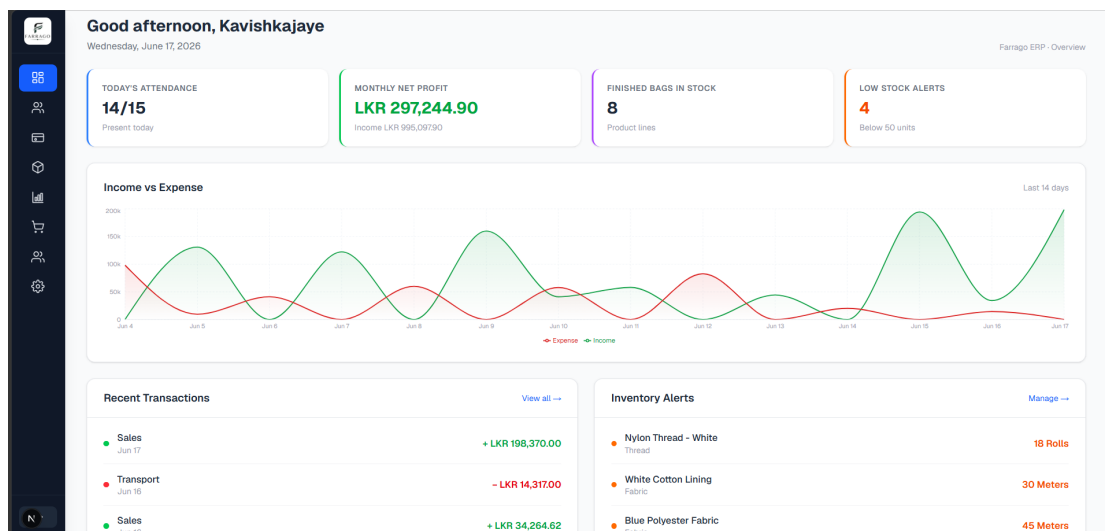


Figure 3.14: The consolidated dashboard.

4 External Interface Requirements

This chapter specifies the interfaces between the system and the people, hardware and software in its environment.

4.1 User Interfaces

The system presents a single-page web application rendered in the browser. A persistent left navigation rail provides access to every module; it is collapsible to an icon-only rail and expands on demand to conserve screen space.

Table 4.1: Principal screens.

Screen	Purpose
Sign-in	Authenticate the user.
Dashboard	Headline figures, income/expense trend, recent activity and alerts.
HR	Employee register and employee add/edit.
Attendance	Daily and bulk attendance and piece capture.
Payroll	Period calculation, adjustments, mark-as-paid, payslip / master report.
Inventory	Item lists by class, add/edit, restock.
Suppliers	Supplier register.
Accounting	Transactions, period summary, profit & loss report.
Sales	Invoice creation, analytics charts, sales report.
Customers	Customer registry.
Settings	Piece-rate configuration.

The following user-interface conventions apply throughout:

- **UI-1** All data-entry forms shall validate input and display a clear message on rejection.
- **UI-2** Destructive actions (delete) shall require confirmation.
- **UI-3** Monetary values shall be shown in LKR with thousands separators and two decimals.
- **UI-4** Long-running actions shall show a loading state, and lists shall show an empty state when no data exists.

- **UI-5** Generated reports shall open as downloadable PDF files.

4.2 Hardware Interfaces

The system has no direct hardware interfaces. It requires only:

- **HW-1** A client device (desktop or laptop) capable of running a modern web browser, with a network connection.
- **HW-2** A server host able to run a Node.js process and reach the database and identity provider over the network.

4.3 Software Interfaces

Table 4.2: Software interfaces.

Interface	Description
Firebase Authentication	Identity provider. The client authenticates users and obtains an ID token; the API verifies the token via the Firebase Admin SDK and reads the role custom claim.
MongoDB (via Mongoose)	Document database storing all persistent collections. Accessed by the API through the Mongoose ODM.
Web client runtime	Next.js / React application served to the browser; charts rendered with a client charting library; PDF reports generated client-side.
Hosting platforms	The reference deployment serves the client from a static/web host and the API from a Node.js host; the database is hosted on MongoDB Atlas.

4.4 Communication Interfaces

- **COM-1** The client and server shall communicate over HTTPS using a JSON REST API.
- **COM-2** Every protected request shall carry the Firebase ID token in an `Authorization: Bearer <token>` header.
- **COM-3** Responses shall use a consistent envelope (`success`, `data/message`) and conventional HTTP status codes (200, 201, 400, 401, 403, 404, 409, 500).
- **COM-4** Cross-origin requests from the web client's origin shall be permitted (CORS).

4.4.1 REST API Endpoint Catalogue

The table below catalogues the API. Paths are relative to the `/api` base. The *Roles* column lists the roles permitted (the **ADMIN** role satisfies every check implicitly).

Table 4.3: REST API endpoint catalogue.

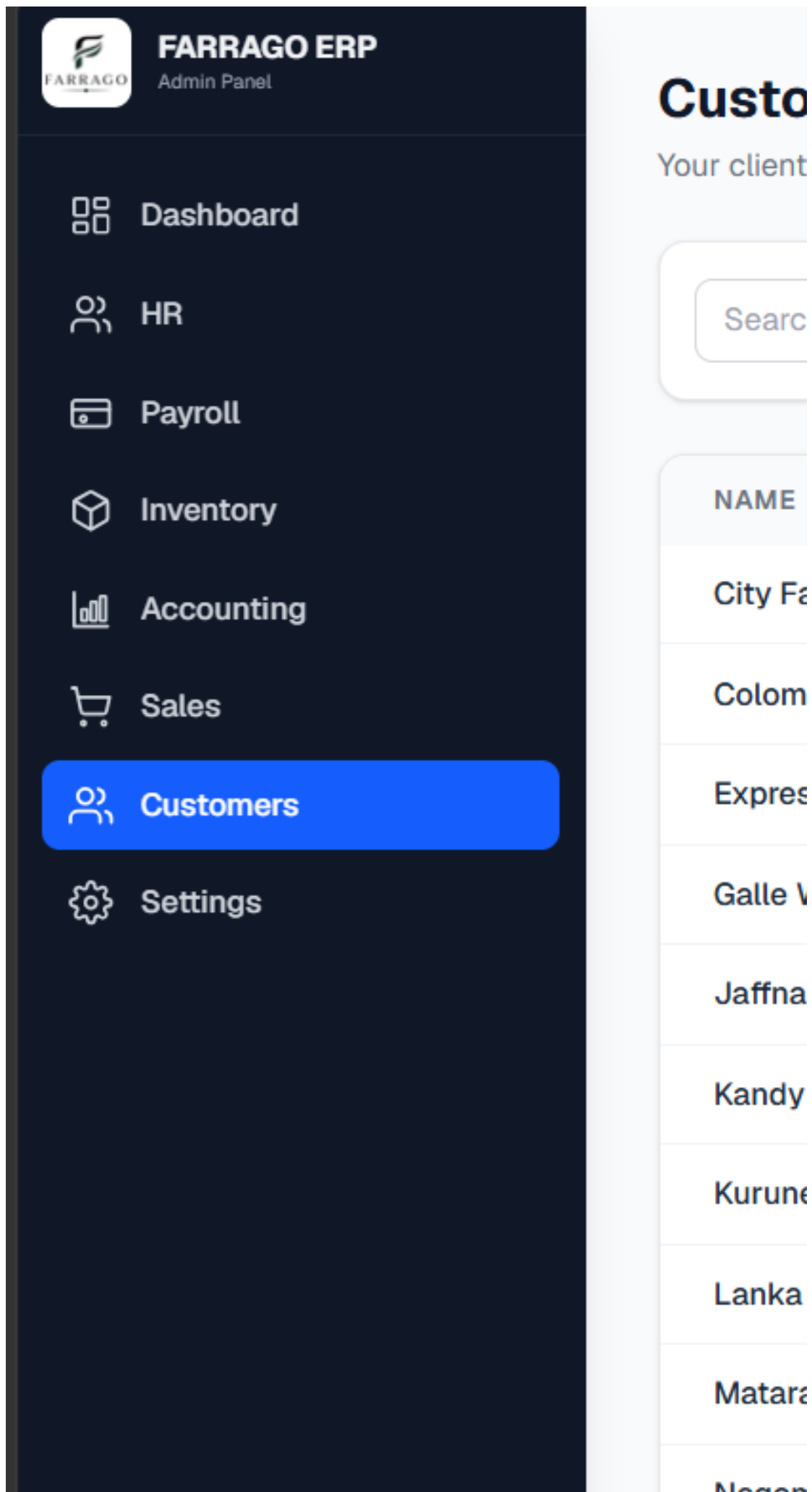
Method	Path	Purpose / Roles
GET	/health	Service liveness check. Public.
<i>Authentication is required for all routes below.</i>		
POST	/employees	Create employee. HR roles.
PUT	/employees/:id	Update employee. HR roles.
GET	/employees	List employees. HR + attendance roles.
GET	/employees/:id	Get one employee. HR roles.
GET	/employees/:id/salary	Monthly salary calculation. HR roles.
POST	/attendance	Log attendance. Attendance roles.
GET	/attendance	List attendance. Attendance roles.
GET	/attendance/history/:employeeId	Per-employee history. HR roles.
GET	/piece-tasks	List piece tasks. Attendance roles.
POST	/piece-tasks	Create piece task. HR roles.
PUT	/piece-tasks/:id	Update piece task. HR roles.
DELETE	/piece-tasks/:id	Delete piece task. Admin, Manager.
GET	/payroll/calculate	Calculate period payroll. HR roles.
GET	/payroll/adjustments	List adjustments. HR roles.
POST	/payroll/adjustments	Create adjustment. HR roles.
DELETE	/payroll/adjustments/:id	Delete adjustment. Admin, Manager.
PUT	/payroll/status	Mark period paid/hold/pending. Admin, Manager.
GET	/inventory-items	List items. Inventory read roles.
GET	/inventory-items/:id	Get one item. Inventory read roles.
GET	/inventory-items/meta/options	Type/category options. Inventory read.
POST	/inventory-items	Create item. Inventory write roles.
PATCH	/inventory-items/:id/restock	Restock item. Inventory write roles.
PUT	/inventory-items/:id	Update item. Inventory write roles.
DELETE	/inventory-items/:id	Delete item. Inventory write roles.

Table 4.3: REST API endpoint catalogue.

Method	Path	Purpose / Roles
GET	/suppliers	List suppliers. Inventory read roles.
GET	/suppliers/:id	Get one supplier. Inventory read roles.
POST	/suppliers	Create supplier. Inventory write roles.
PUT	/suppliers/:id	Update supplier. Inventory write roles.
DELETE	/suppliers/:id	Delete supplier. Inventory write roles.
POST	/transactions	Create transaction. Accounting roles.
GET	/transactions	List transactions. Accounting roles.
DELETE	/transactions/:id	Delete transaction. Accounting roles.
GET	/accounting/summary	Profit & loss summary. Accounting roles.
GET	/accounting/meta	Category/method options. Accounting roles.
POST	/sales	Create invoice. Sales roles.
GET	/sales	List invoices. Sales roles.
GET	/sales/:id	Get one invoice. Sales roles.
PATCH	/sales/:id/payment	Record a payment. Sales roles.
PUT	/sales/:invoiceNo/status	Settle invoice. Sales roles.
GET	/sales/summary	Sales KPI summary. Sales roles.
GET	/sales/chart-data	Daily sales trend. Sales roles.
GET	/sales/top-customers	Top customers. Sales roles.
GET	/sales/top-products	Top products. Sales roles.
GET	/sales/monthly-revenue	Monthly revenue. Sales roles.
GET	/customers	List/search customers. Sales roles.
GET	/customers/:id	Get one customer. Sales roles.
POST	/customers	Create customer. Sales roles.
PUT	/customers/:id	Update customer. Sales roles.
DELETE	/customers/:id	Delete customer. Sales roles.
GET	/dashboard/summary	Consolidated dashboard data. HR roles.
POST	/approvals	Submit payroll for approval. HR roles.
GET	/approvals	List approval requests. HR roles.

Table 4.3: REST API endpoint catalogue.

Method	Path	Purpose / Roles
PUT	/approvals/:id/approve	Approve & settle payroll. Admin.
PUT	/approvals/:id/reject	Reject the request. Admin.
GET	/users	List users & roles. Admin.
POST	/users	Create a user. Admin.
PUT	/users/:uid/role	Change a user's role. Admin.
DELETE	/users/:uid	Delete a user. Admin.
GET	/notifications	Admin activity feed. Admin.
PUT	/notifications/read	Mark all read. Admin.



5 Non-Functional Requirements

This chapter specifies the quality attributes the system must satisfy. Each requirement is identified as **NFR-*n***.

5.1 Performance Requirements

- **NFR-1** Interactive screens shall load their data within 3 seconds under normal conditions (excluding cold-start of a sleeping host).
- **NFR-2** List and summary queries shall remain responsive for the expected single-factory data volume (thousands of employees, transactions and invoices), aided by indexed queries.
- **NFR-3** Report generation (PDF) shall complete within a few seconds for typical report sizes and shall not block the user interface.
- **NFR-4** The system shall support concurrent use by all factory administrative staff (on the order of tens of simultaneous users) without functional degradation.

5.2 Security Requirements

- **NFR-5** All client-server communication shall use HTTPS.
- **NFR-6** Every protected endpoint shall require a valid Firebase ID token; the server shall verify the token's signature and expiry on each request.
- **NFR-7** Authorization shall be enforced on the server by role, and shall not rely on client-side checks alone.
- **NFR-8** The system shall not store user passwords; credential management is delegated to Firebase Authentication.
- **NFR-9** Database credentials and service-account keys shall be supplied through environment configuration and shall never be committed to source control.
- **NFR-10** The system shall apply least privilege: each role is granted only the operations its duties require, and sensitive financial functions (accounting, sales, payroll approval) are restricted to **ADMIN** and **MANAGER**.

- **NFR-11** Input shall be validated server-side; invalid or malformed requests shall be rejected with an appropriate error rather than causing undefined behaviour.

5.3 Reliability and Availability

- **NFR-12** Cross-module side-effects shall be best-effort and shall not compromise the integrity of the primary record: a failure to post a derived accounting or inventory record shall not roll back the primary action, and shall be reported as a non-fatal warning.
- **NFR-13** Stock quantities shall never become negative as a result of concurrent operations.
- **NFR-14** Payroll posting shall be idempotent so that repeated mark-as-paid actions cannot create duplicate expenses.
- **NFR-15** A failure in one dashboard aggregation shall not blank the entire dashboard.

5.4 Usability

- **NFR-16** The interface shall be operable by staff with basic computer literacy, using clear labels, consistent layouts and plain-language validation messages.
- **NFR-17** Navigation to any module shall be reachable within two clicks from the dashboard.
- **NFR-18** Reports shall be presented in a clean, print-ready, monochrome layout suitable for filing and sharing.

5.5 Maintainability and Portability

- **NFR-19** The system shall follow a modular separation of concerns (routes, controllers, models, services) so that a module can be modified with minimal impact on others.
- **NFR-20** Cross-module posting logic shall be centralised in a shared service to avoid duplication.
- **NFR-21** The system shall be deployable to standard Node.js and MongoDB hosting without code changes, configured through environment variables.
- **NFR-22** Date/time bucketing for reports shall use a configurable business timezone (default Asia/Colombo).

5.6 Scalability

- **NFR-23** The API shall be stateless so that it can be scaled horizontally behind a load balancer if demand grows.
- **NFR-24** The data model shall accommodate growth in transactional data through indexing and bounded, paginated queries.

6 System Models

This chapter presents the supporting models for the system: the use-case model, the data model, a representative sequence model and the deployment architecture. Diagram slots show a placeholder until the corresponding image is added to the `images/` folder.

6.1 Use-Case Model

The actors of the system are the five roles described in Section 2.3. Table 6.1 summarises the principal use cases and the actors that initiate them.

Table 6.1: Principal use cases and their actors.

Use case	Primary actor(s)
Sign in / sign out	All roles
Manage employees	Admin, Manager, Supervisor
Record attendance & pieces	Admin, Manager, Supervisor, Maintainer
Configure piece rates	Admin, Manager, Supervisor
Calculate payroll	Admin, Manager, Supervisor
Manage payroll adjustments	Admin, Manager, Supervisor
Approve / mark payroll paid	Admin, Manager
Generate payslip / master payroll	Admin, Manager, Supervisor
Manage inventory & restock	Admin, Manager, Supervisor
Manage suppliers	Admin, Manager, Supervisor
Record transactions	Admin, Manager
Generate profit & loss report	Admin, Manager
Create invoice & record payments	Admin, Manager
Manage customers	Admin, Manager
Generate sales report	Admin, Manager
View dashboard	Admin, Manager, Supervisor

6.2 Data Model (Entity-Relationship)

The system persists the following collections. Field-level definitions are given in Appendix A.

Table 6.2: Data collections.

Collection	Role in the system
Employee	Master record of each worker, including wage scheme.
PieceTask	Defined production task and its rate per piece.
Attendance	A worker’s status for a date, with captured piece quantities.
PayrollAdjustment	A monthly allowance, bonus or deduction for an employee.
Supplier	A raw-material supplier.
InventoryItem	A stock item (raw material, finished good or defective).
Customer	A sales client.
Sale	A customer invoice with embedded line items.
Transaction	An income or expense accounting entry.

The principal relationships are:

- An **Attendance** record references one **Employee** and may reference one or more **PieceTasks** through its captured pieces.
- A **PayrollAdjustment** references one **Employee**.
- An **InventoryItem** may reference one **Supplier**.
- A **Sale** references a **Customer** (or stores a walk-in snapshot) and embeds its line items, which correspond to finished-good **InventoryItems** by name.
- A **Transaction** may reference the record that produced it (a **Sale**, an **InventoryItem** purchase, or a payroll batch) through its source and reference fields.

6.3 Sequence Model: Create Sale

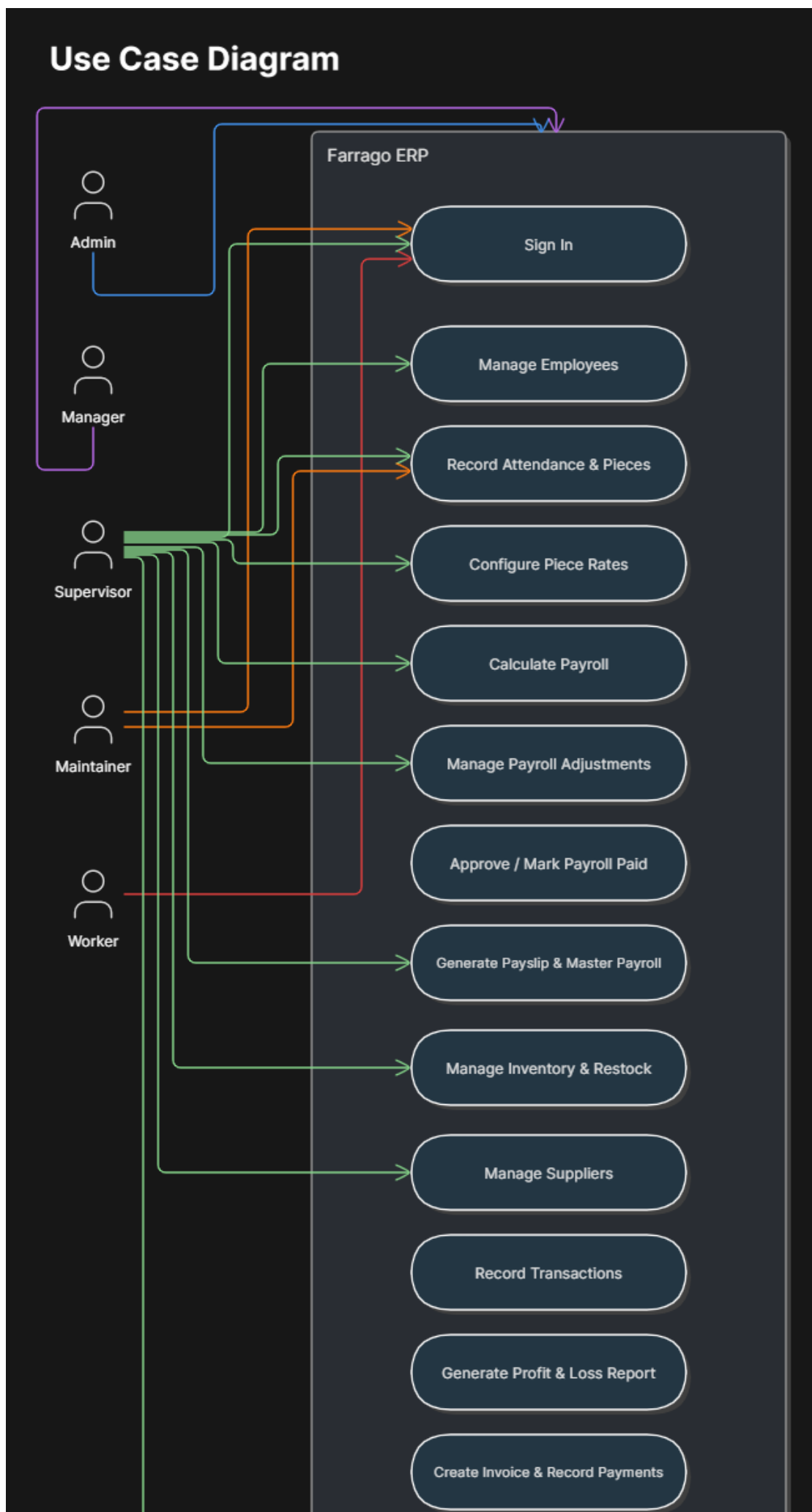
The “Create Sale” interaction exercises the system’s defining cross-module behaviour. Its steps are:

1. The user submits a new invoice (customer, line items, discount, tax, amount paid).
2. The API validates the request and computes totals server-side.
3. The API persists the **Sale**.

4. The API decrements matching finished-good **InventoryItem** stock (floored at zero), collecting any warnings.
5. The API posts an income **Transaction** for the grand total.
6. The API returns the saved invoice together with any non-fatal warnings.

6.4 Deployment Architecture

The system is deployed as three cooperating tiers together with two managed external services: the web client (browser), the REST API (Node.js host), the database (MongoDB Atlas), and Firebase Authentication for identity. The client calls the API over HTTPS; the API verifies tokens with Firebase and reads/writes the database.



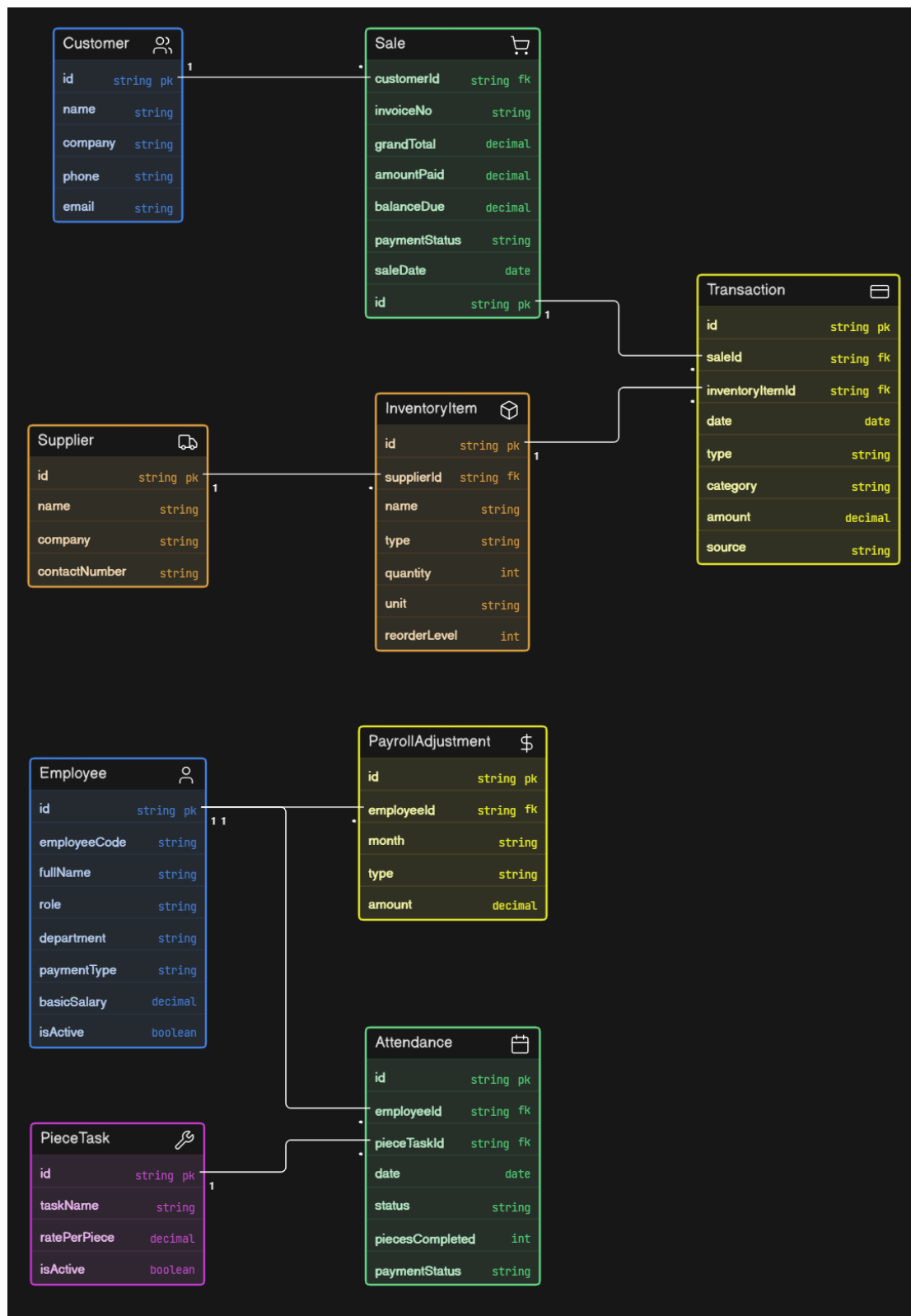


Figure 6.2: Entity-relationship diagram.

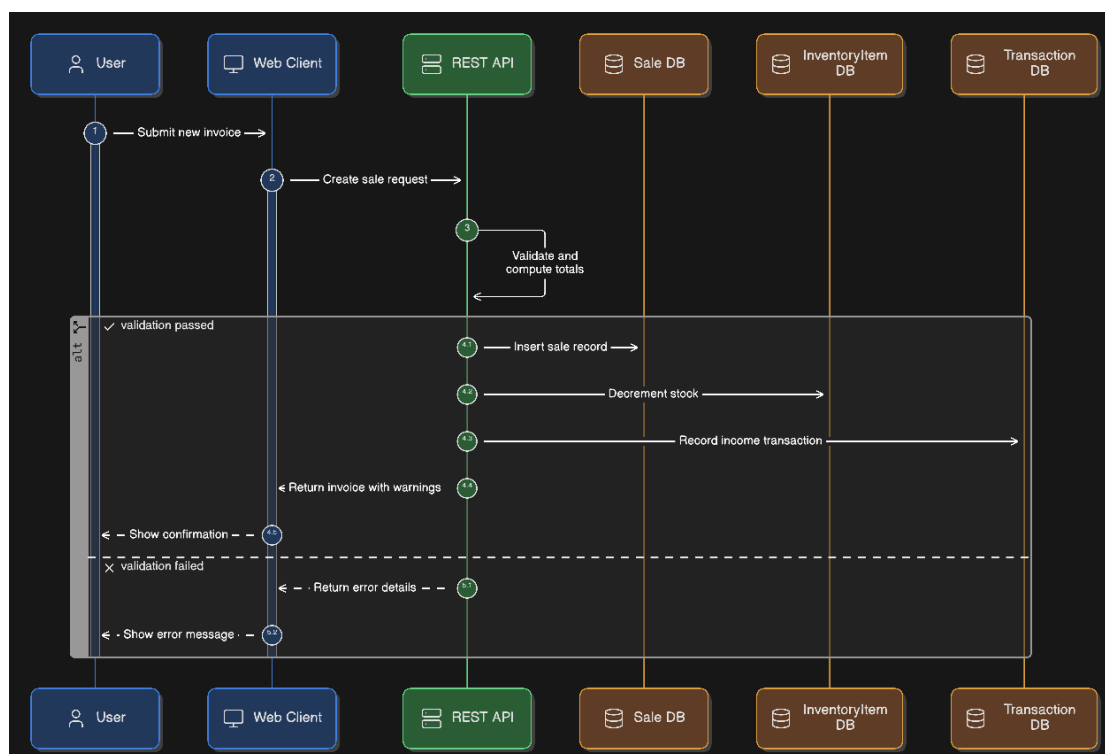


Figure 6.3: Sequence diagram — Create Sale.

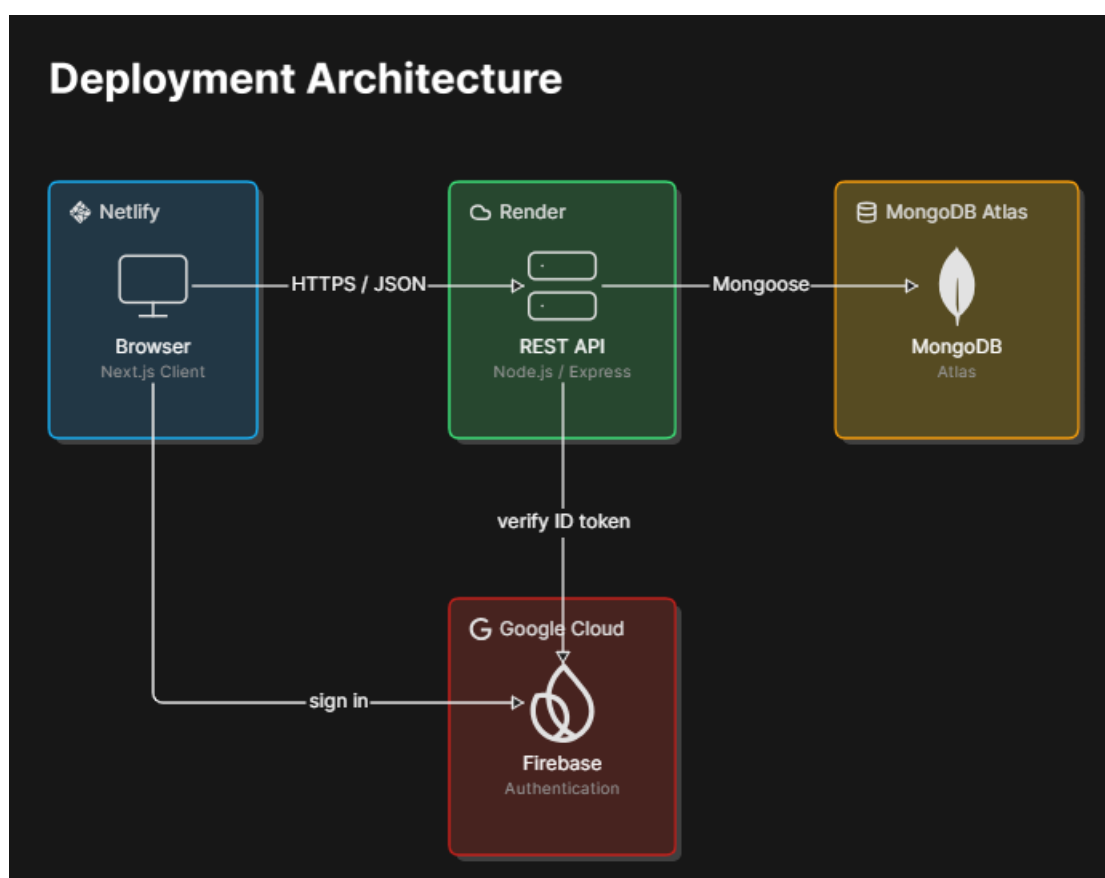


Figure 6.4: Deployment architecture.

A Data Dictionary

This appendix defines the fields of each persistent collection. Reference fields (suffixed conceptually by “ref”) hold the identifier of a record in another collection. Audit fields such as creation and update timestamps are maintained automatically and are omitted for brevity.

A.1 Employee

Table A.1: Employee fields.

Field	Type	Description
employeeCode	String	Unique human-readable employee identifier.
fullName	String	Employee’s full name.
nic	String	National identity card number (optional).
contactNumber	String	Telephone number (optional).
role	String	Job role / designation.
department	String	Department the employee belongs to.
payment.type	Enum	Wage scheme: PIECE_RATE or FIXED_BASIC.
payment.basicSalary	Number	Monthly basic salary (fixed-basic only).
isActive	Boolean	Whether the employee is currently employed.

A.2 PieceTask

Table A.2: PieceTask fields.

Field	Type	Description
taskName	String	Name of the production task.
ratePerPiece	Number	Amount paid per completed unit.
description	String	Optional description of the task.
isActive	Boolean	Whether the task is currently in use.

A.3 Attendance

Table A.3: Attendance fields.

Field	Type	Description
employee	Ref	The employee the record belongs to.
date	Date	Calendar day of the record.
status	Enum	PRESENT, HALF_DAY, ABSENT or LEAVE.
pieces	Array	Per-task piece counts: { pieceTaskId (ref), quantity }.
paymentStatus	Enum	PENDING, PAID or HOLD (payroll lock state).
paidAt	Date	When the record was marked paid (audit).
paidBy	String	Who marked the record paid (audit).
notes	String	Optional notes.

A.4 PayrollAdjustment

Table A.4: PayrollAdjustment fields.

Field	Type	Description
employeeId	Ref	The employee the adjustment applies to.
month	String	Month of effect, formatted YYYY-MM.
type	Enum	ALLOWANCE, BONUS or DEDUCTION.
amount	Number	Positive monetary amount of the adjustment.
description	String	Optional reason / note.
createdBy	String	Who created the adjustment (audit).

A.5 Supplier

Table A.5: Supplier fields.

Field	Type	Description
name	String	Supplier (contact) name.

Table A.5: Supplier fields.

Field	Type	Description
company	String	Company name (optional).
contactNumber	String	Telephone number (optional).
address	String	Postal address (optional).
isActive	Boolean	Whether the supplier is active.

A.6 InventoryItem

Table A.6: InventoryItem fields.

Field	Type	Description
name	String	Item name.
type	Enum	RAW_MATERIAL, FINISHED_GOOD or DEFECTIVE.
category	String	Optional sub-category.
quantity	Number	Current quantity on hand (non-negative).
unit	String	Unit of measure (e.g. Pieces).
reorderLevel	Number	Threshold for low-stock flagging.
supplierId	Ref	Linked supplier (optional).
notes	String	Optional notes.
isActive	Boolean	Whether the item is active.

A.7 Customer

Table A.7: Customer fields.

Field	Type	Description
name	String	Customer name.
company	String	Company name (optional).
phone	String	Telephone number (optional).
email	String	Email address (optional).
address	String	Postal address (optional).
notes	String	Optional notes.
isActive	Boolean	Whether the customer is active.

A.8 Sale

Table A.8: Sale fields.

Field	Type	Description
invoiceNo	String	Unique invoice number.
customer	Object	Snapshot { name, phone, address } of the buyer.
items	Array	Line items: { description, sku, quantity, unitPrice, lineTotal }.
subTotal	Number	Sum of line totals.
discount	Number	Discount applied to the subtotal.
taxRate	Number	Tax percentage applied.
taxAmount	Number	Computed tax amount.
grandTotal	Number	Final payable amount.
amountPaid	Number	Total amount paid so far.
balanceDue	Number	Outstanding balance.
paymentStatus	Enum	UNPAID, PARTIAL or PAID.
saleDate	Date	Date of the sale.
createdBy	Ref	Employee who created the invoice (optional).

A.9 Transaction

Table A.9: Transaction fields.

Field	Type	Description
date	Date	Date of the transaction.
type	Enum	INCOME or EXPENSE.
category	String	Category (e.g. Sales, Payroll, Raw Materials).
amount	Number	Positive monetary amount.
paymentMethod	Enum	CASH, BANK_TRANSFER or CHEQUE.
source	Enum	MANUAL, SALE, PAYROLL or PURCHASE (origin).
referenceId	Ref	Identifier of the originating record (optional).

Table A.9: Transaction fields.

Field	Type	Description
referenceModel	String	Collection the reference points to (optional).
referenceKey	String	Idempotency / grouping tag (optional).
notes	String	Optional notes.
createdBy	String	Who created the entry (audit).

B Glossary

Table B.1: Glossary of domain terms.

Term	Meaning
Effective day	Productive day-equivalent: a present day counts as 1, a half-day as 0.5.
Grand total	The final payable amount on an invoice after discount and tax.
Idempotent	An operation that has the same effect whether performed once or repeated.
Lock (payroll)	Marking attendance for a period as paid so it can no longer be altered casually.
Low stock	An item at or below its reorder level.
Master payroll	A factory-wide payroll summary report for a period.
Net payable	An employee's take-home wage after adjustments.
Payment status (invoice)	Unpaid, partial or paid, derived from amount paid versus grand total.
Reorder level	Stock threshold that triggers a low-stock flag.
Source (transaction)	Where a transaction originated: manual or an automatic posting from sales, payroll or procurement.
Walk-in customer	A buyer entered directly on an invoice without a saved customer record.

C Proposed Future Enhancements

The following are candidate enhancements beyond the current scope (Section 1.2):

- Integration with tax / government e-filing systems.
- An online, customer-facing ordering portal.
- Bank or payment-gateway integration for collections.
- Multi-factory / multi-branch consolidation and reporting.
- A native mobile application for floor supervisors.
- Barcode / QR-based stock and finished-goods tracking.
- Role-aware analytics and configurable management reports.