Software Requirements Specification

Project: Finest Meat BD – Grading, Packaging, and Transport Management System

1. Introduction

1.1 Purpose

The purpose of this system is to provide a web-based platform to **grade, package, and manage transport of meat-based products**. It ensures transparency, traceability, and efficiency in the supply chain. This document specifies the requirements for system design, development, and deployment.

1.2 Document Conventions

- **Abbreviations:**
  - **QC**: Quality Control
  - **CRUD**: Create, Read, Update, Delete
  - **DBMS**: Database Management System
- **Standards:**
  - IEEE SRS formatting guidelines
  - PHP 8.0+ coding standards

1.3 Intended Audience and Reading Suggestions

- **Developers & Testers:** Focus on **Sections 3 & 4** for functional and interface requirements.
- **Project Managers:** Review **Sections 1 & 2** for project scope and overall description.
- **Stakeholders (faculty, clients):** Review **Section 2** for features and user expectations.

1.4 Project Scope

Finest Meat BD is a **centralized supply chain management platform** for meat industries. It standardizes meat grading, tracks packaging batches, and manages transportation

logistics.
Objectives:

- Ensure **food quality** and **traceability**.
- Streamline **packaging and batch records**.
- Provide **transport monitoring**.
- Enhance **decision-making** with dashboards and reports.

1.5 References

- IEEE SRS Template
- CSE 309 Lecture Notes on Requirements Engineering
- GitHub Project Repository: Finest Meat BD

---

2. Overall Description

2.1 Product Perspective

- Standalone web application.
- Three-tier architecture: **Frontend (HTML/CSS/JS)**, **Backend (PHP)**, **Database (MySQL)**.
- Extensible to integrate IoT sensors in future.

2.2 Product Features

- **Authentication & Role Management** (Admin, QC Inspector, Packaging Staff, Transport Manager).
- **Meat Grading**: Record and validate quality attributes.
- **Packaging**: Create batches, track inventory, label products.
- **Transport Management**: Schedule vehicles, assign routes, track deliveries.
- **Reports**: Generate/export grading, packaging, and transport summaries (PDF/Excel).

2.3 User Classes and Characteristics

- **Admin:** Manage users, system settings.
- **QC Inspector:** Perform grading and approve quality.
- **Packaging Staff:** Handle batch assignments and inventory updates.
- **Transport Manager:** Plan and monitor logistics.
- **Buyer (future scope):** View product details and track orders.

2.4 Operating Environment

- **Server:** Apache (XAMPP/Laragon).
- **Database:** MySQL 8.0.
- **Client:** Works in Chrome, Firefox, Edge.
- **OS:** Windows/Linux.

2.5 Design and Implementation Constraints

- Must use PHP + MySQL stack.
- Deployed via localhost or shared hosting.
- Limited to course deadlines and resources.

2.6 User Documentation

- PDF User Manual (for Admin and Staff).
- Inline help texts within dashboard forms.

2.7 Assumptions and Dependencies

- Users have internet access.
- Accurate grading depends on human inspectors.
- Transport management depends on available vehicles.

---

3. System Features

3.1 User Authentication

- **Description and Priority:** Secure login system. High priority.

- **Stimulus/Response:** Users log in → Role-based dashboard displayed.
- **Functional Requirements:**
  - FR1: System shall allow registration and login.
  - FR2: System shall enforce role-based access.
  - FR3: Passwords stored using hashing.

3.2 Meat Grading

- **Description and Priority:** Core functionality. High priority.
- **Stimulus/Response:** QC enters grading info → Stored in DB.
- **Functional Requirements:**
  - FR4: Record product grade, weight, freshness.
  - FR5: Generate grading reports.

3.3 Packaging Management

- **Description and Priority:** Organizes batches. High priority.
- **Stimulus/Response:** Staff assign products → System updates inventory.
- **Functional Requirements:**
  - FR6: Assign products to packaging batches.
  - FR7: Generate packaging records.

3.4 Transport Management

- **Description and Priority:** Handles logistics. High priority.
- **Stimulus/Response:** Manager schedules transport → Stored & displayed in dashboard.
- **Functional Requirements:**
  - FR8: Schedule vehicle and driver.
  - FR9: Log delivery completion.

3.5 Reports & Export

- **Description and Priority:** Essential for audits. Medium priority.

- **Stimulus/Response:** Admin requests report → PDF/Excel generated.
- **Functional Requirements:**
  - FR10: Export grading reports.
  - FR11: Export transport logs.

---

4. External Interface Requirements

4.1 User Interfaces

- Dashboard with navigation sidebar.
- Forms for CRUD operations.
- Export buttons (PDF/Excel).

4.2 Hardware Interfaces

- Standard PCs/laptops.
- (Future) Barcode scanner for packaging.

4.3 Software Interfaces

- PHP-MySQL connection (mysqli/PDO).
- FPDF library for reports.

4.4 Communications Interfaces

- HTTPS for secure deployment.
- Localhost testing via HTTP.

---

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Handle 100+ concurrent users.
- Response time ≤ 2 seconds for CRUD operations.

5.2 Safety Requirements

- Daily database backups.
- Graceful error handling (no data loss).

5.3 Security Requirements

- Prepared statements (SQL injection protection).
- Session management (logout on timeout).
- Escaping user input (XSS prevention).

5.4 Software Quality Attributes

- **Usability:** Simple UI with consistent layout.
- **Maintainability:** Modular PHP code.
- **Portability:** Runs on any PHP/MySQL-supported server.

---

6. Other Requirements

- Future integration with IoT meat quality sensors.
- Support for multilingual UI (English/Bangla).

Appendices

A. Glossary

- **Batch:** Group of packaged meat products with same production date.
- **Traceability:** Ability to track product lifecycle from grading to delivery.

B. Analysis Models

- Use Case Diagram: Shows actors (Admin, QC, Staff, Transport Manager) and modules.
- ER Diagram: Database schema (Users, Products, Batches, Transport, Reports).

C. Issues List

- Pending: Integration with IoT devices.
- Resolved: Database schema design (as per SQL files).
