目錄

Abstract

This report documents the development of a comprehensive web-based Fruits Management System for Acer International Bakery (AIB), an international bakery chain operating in Japan, USA, and Hong Kong. The system addresses AIB's need for efficient cross-border materials management by enabling real-time tracking of fruit inventory, facilitating fruit borrowing between local shops, supporting advance reservations from source countries, and streamlining the delivery process. Developed using Java EE with JSP/Servlets and following the MVC architecture, the system provides role-specific interfaces for bakery shop staff, warehouse personnel, and senior management. The implementation showcases effective application of enterprise systems concepts to solve real-world supply chain challenges in a multi-national retail environment.

Introduction

Company Background

Acer International Bakery (AIB) is a multinational bakery chain with operations across major cities in Japan, USA, and Hong Kong. As its name suggests, AIB specializes in international-style bakery products that feature fruits as primary ingredients. The company maintains a complex supply chain where fruits are sourced from specific origin countries and distributed to bakery shops across different international locations. This global operation requires efficient logistics management to ensure that quality ingredients are available at each bakery location when needed.

Project Background

AIB currently faces challenges with its materials management process, particularly in coordinating fruit deliveries from source countries to bakery shops across international borders. The existing manual system lacks real-time visibility of inventory levels, creates inefficiencies in the reservation process, and makes it difficult to coordinate borrowing between shops in the same city. This has resulted in supply shortages, excess inventory costs, and suboptimal resource allocation.

To address these issues, AIB has commissioned the development of a web-based Fruits Management System that will digitize and streamline their entire fruit supply chain. The system aims to provide real-time inventory visibility, facilitate efficient borrowing between shops, enable advance reservations, and support analytical decision-making for management.

The Scope and Overview of This Project

The Fruits Management System encompasses the following key areas:

1. **Inventory Management**: Real-time tracking of fruit stock levels across all locations (shops, city warehouses, and country warehouses) with source country information.
2. **Inter-Shop Borrowing**: Facilitating fruit borrowing between shops in the same city to address immediate shortages.
3. **Advance Reservation**: Enabling shops to reserve fruits up to 14 days in advance from source countries.
4. **Delivery Coordination**: Tracking fruit movements from source warehouses to country central warehouses and finally to local shops.
5. **Analytics and Reporting**: Providing management with insights into consumption patterns and reservation needs.
6. **User Management**: Supporting different user roles with appropriate access controls.

The project delivers a responsive web application built with Java EE technology, implementing JSP/Servlets, JavaBeans, JDBC, and custom tags following the MVC architecture pattern. The system features role-specific interfaces for bakery staff, warehouse personnel, and senior management with appropriate authentication and authorization controls.

Assumptions and User/System Requirements

Business Process Assumptions

1. AIB operates a three-tier supply chain: source country warehouses → destination country central warehouses → local bakery shops.
2. Fruits are sourced from specific countries based on quality and availability (e.g., certain fruits may only be available from specific origin countries).
3. Shops can borrow fruits only from other shops within the same city.
4. Fruit reservations must be made at least 14 days in advance to allow for international shipping logistics.
5. Central warehouses in each country aggregate reservation requests before placing orders with source country warehouses.
6. Fruits are measured and tracked in kilograms (kg) across all locations in the system.

User Role Assumptions

1. **Bakery Staff** are responsible for managing inventory at individual shops, making fruit reservations, and handling local borrowing.
2. **Warehouse Staff** manage inventory at central warehouses, approve reservation requests, and coordinate deliveries.
3. **Senior Management** require analytics for decision-making but do not handle day-to-day operations.
4. Each user has access only to functionality relevant to their role and location.

Technical Environment Assumptions

1. All locations have reliable internet connectivity to access the centralized web-based system.
2. The system will operate on standard web browsers across desktop and mobile devices.
3. Database operations will be centralized but support distributed access with appropriate security controls.

4. System performance must accommodate peak usage during inventory updates and end-of-day operations.
5. Data synchronization may experience brief delays but should complete within 5 minutes across all locations.

Data Management Assumptions

1. Each fruit has specific attributes including name, category, source country, and minimum threshold quantities.
2. Historical data for stock movements, borrowings, and reservations must be maintained for reporting and auditing.
3. Seasonal variations in fruit availability and consumption will be tracked for forecasting.
4. Stock levels have status indicators (normal, low, critical) based on predefined thresholds.

User Requirements

Bakery Shop Staff Requirements

1. **Inventory Management**
   - View real-time fruit inventory with visual status indicators and source country information
   - Filter fruits by type, source country, and stock status
   - Update stock levels when receiving or using fruits
   - Track stock history for audit and reconciliation purposes
2. **Borrowing Management**
   - Request to borrow fruits from other shops in the same city
   - View available quantities at nearby shops
   - Track status of borrowing requests (pending, approved, rejected)
   - Record returns of borrowed fruits
3. **Reservation Management**
   - Reserve specific fruit quantities from source countries up to 14 days in advance
   - View status of pending reservations
   - Receive notifications when reserved fruits are delivered
   - Cancel or modify reservations before approval

4. **Delivery Tracking**
    - Monitor status of incoming fruit deliveries (from reservations or borrowing)
    - Confirm receipt of deliveries and update inventory accordingly
    - Report discrepancies in delivered quantities

Warehouse Staff Requirements

1. **Inventory Control**
    - Maintain accurate stock records at central warehouses
    - Process check-ins from source countries
    - Manage check-outs to local bakery shops
    - Conduct inventory reconciliation and adjustments
2. **Reservation Processing**
    - Review and approve reservation requests from bakery shops
    - Aggregate reservation needs by country and fruit type
    - Generate purchase orders for source country warehouses
    - Allocate received stock according to reservation priorities
3. **Delivery Coordination**
    - Schedule deliveries to local bakery shops
    - Generate delivery documentation
    - Track delivery status and confirm completions
    - Manage delivery exceptions and rescheduling

Senior Management Requirements

1. **Analytics Dashboard**
    - View consumption patterns by fruit type, location, and time period
    - Monitor reservation trends and forecast future needs
    - Track inventory efficiency metrics (stock turnover, wastage)
    - Compare performance across different regions
2. **System Administration**
    - Manage user accounts and access permissions
    - Configure system parameters (minimum thresholds, notification settings)
    - Maintain master data (fruit types, locations, suppliers)
    - Perform system audits and security reviews

System Requirements

Functional Requirements

1. **Authentication and Authorization**
   - Secure login system with role-based access control
   - Session management with appropriate timeouts
   - Password encryption and recovery mechanisms
   - Activity logging for security audit
2. **Inventory Management System**
   - Real-time stock level tracking across all locations
   - Automatic status updating based on defined thresholds
   - Stock movement transactions with proper validation
   - Historical tracking with audit trail capabilities
3. **Reservation and Borrowing System**
   - Multi-step approval workflow for reservations
   - Real-time availability checking for borrowing requests
   - Notification system for status updates
   - Conflict resolution for competing requests
4. **Reporting and Analytics**
   - Predefined reports for common operational needs
   - Custom report generation with filtering options
   - Data visualization for trend analysis

Technical Requirements

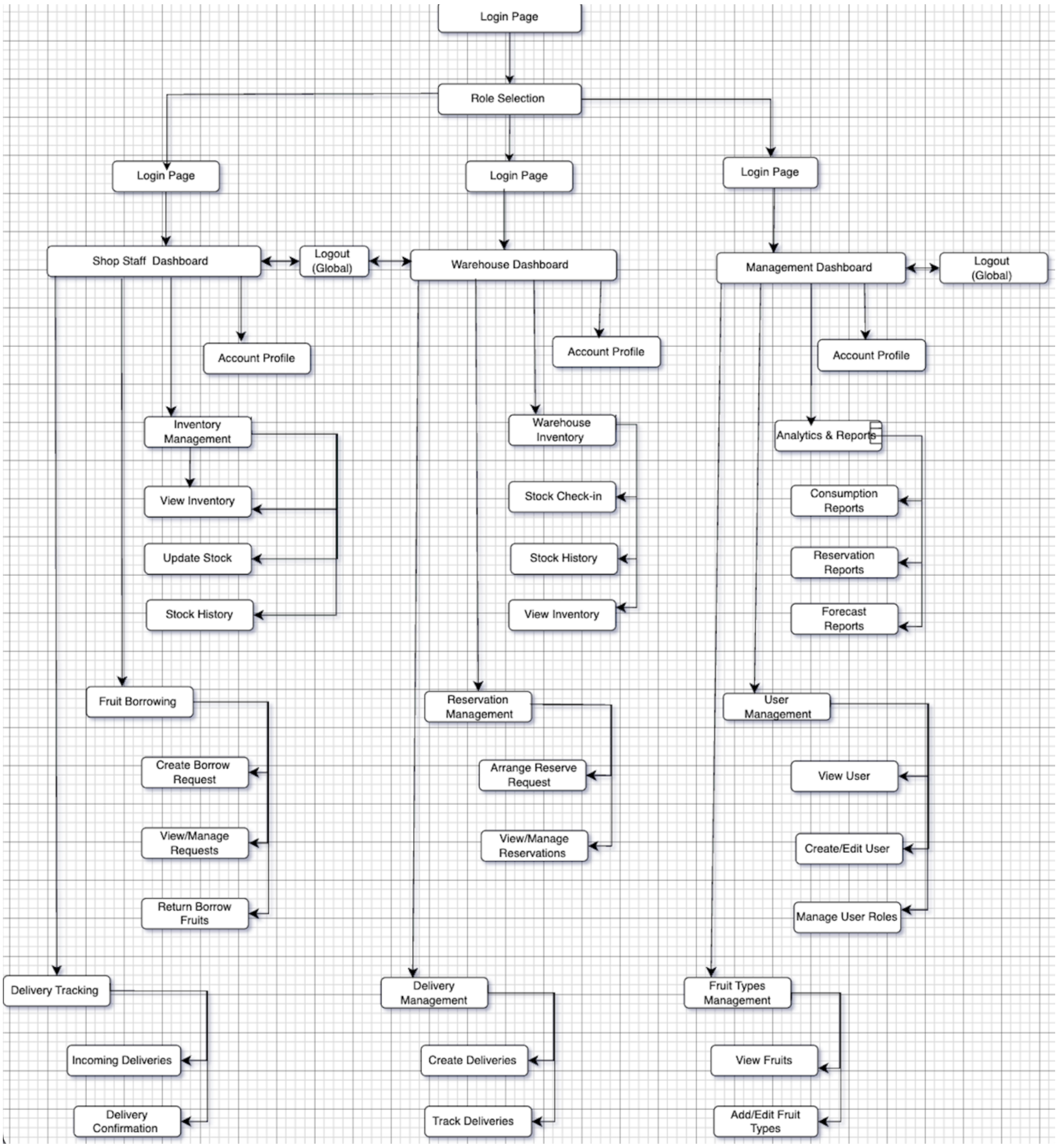
1. **Architecture**
   - MVC pattern implementation using Java EE technologies
   - Modular design allowing for component reuse and maintenance
   - Separation of concerns between presentation, business logic, and data access
2. **Performance**
   - Page load times under 3 seconds for standard operations
   - Support for up to 100 concurrent users
   - Database query optimization for rapid inventory lookups
   - Responsive design for various device form factors
3. **Security**

    - Data encryption for sensitive information
    - Protection against common web vulnerabilities (SQL injection)
    - Secure connection using HTTPS
    - Regular security updates and vulnerability assessments

4. **Reliability**
    - Daily database backups
    - System availability of at least 99.5%
    - Graceful error handling with user-friendly messages
    - Transaction integrity to prevent data corruption

This comprehensive system will transform AIB's fruit supply chain management, enabling more efficient operations, reducing waste, and improving coordination across their international bakery business.

Site Map for AIB Fruits Management System



Site Map Description

Common Access Point

- **Login Page**: Entry point for all users with role-based redirection after authentication

- **Logout Button**: Globally accessible from any authenticated page to securely end the user session

Bakery Shop Staff Section

- **Dashboard**: Overview of shop's fruit inventory status, pending borrowing/reservation requests, and incoming deliveries
- **Inventory Management**:
  - View current inventory with filtering options
  - Update stock levels when using fruits
  - View stock history for auditing
- **Fruit Borrowing**:
  - Create borrowing requests from other shops
  - View and manage existing borrowing requests
  - Record returns of borrowed fruits
- **Reservation Management**:
  - Create reservation requests from source countries
  - View and manage existing reservations
- **Delivery Tracking**:
  - Monitor incoming deliveries from reservations/borrowing
  - Confirm receipt of deliveries

Warehouse Staff Section

- **Dashboard**: Overview of warehouse operations, pending approvals, and delivery statuses
- **Warehouse Inventory**:
  - View current warehouse inventory
  - Process check-ins from source countries
  - View stock history and movements
- **Reservation Management**:
  - Review pending reservation requests
  - Approve/reject reservation requests
  - View aggregated country needs
- **Delivery Management**:
  - Create outgoing deliveries to shops
  - Track delivery status

    - Generate delivery reports

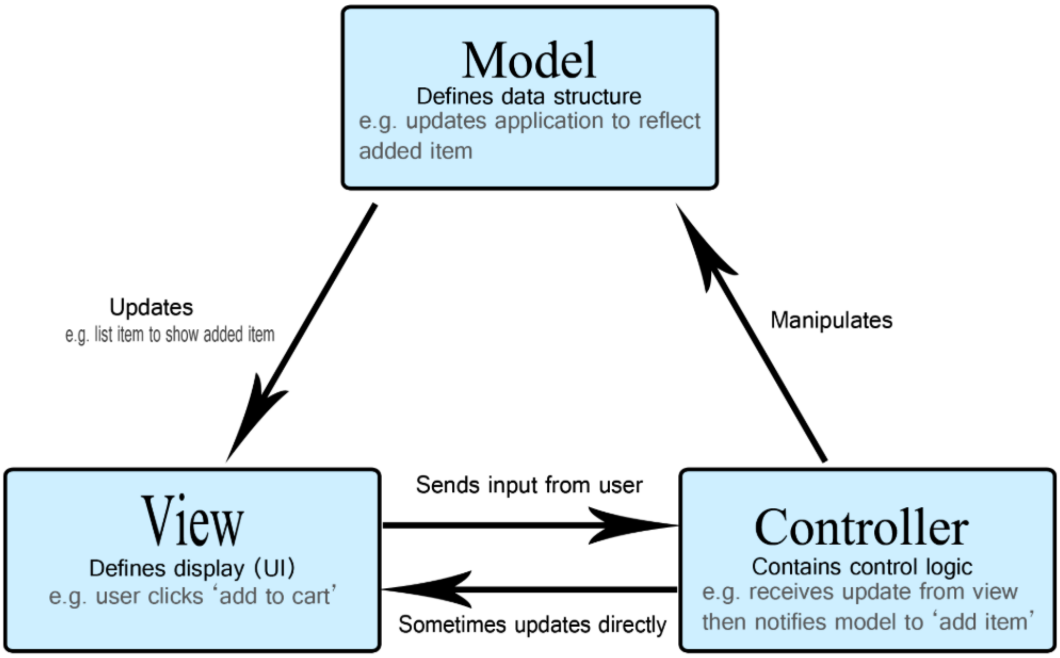
Senior Management Section

- **Dashboard**: Executive overview with key performance metrics and alerts
- **Analytics & Reports**:
    - Consumption reports by location/time period
    - Reservation trends analysis
    - Stock level reports across locations
    - Forecast reports for future planning
- **User Management**:
    - View all system users
    - Create and edit user accounts
    - Manage user roles and permissions
- **Fruit Types Management**:
    - View all fruit types in the system
    - Add/edit fruit types and their attributes

This site map provides a comprehensive overview of the navigational structure of the AIB Fruits Management System, showing how users of different roles can access their required functionality.

System Structure on how MVC Model is Applied

The AIB Fruits Management System implements a robust MVC (Model-View-Controller) architecture to ensure clean separation of concerns, maintainability, and scalability. Based on the comprehensive project file structure observed in the images, here's an in-depth analysis of how the MVC pattern is applied throughout the system:


Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

1. Model Layer (Data and Business Logic)

The Model component represents the application's data structure, business logic, and data access layer. From the project structure, the model layer is organized into two main components:

1.1 JavaBeans (`ict.bean` package)

These classes encapsulate the data and business logic of the application:

- **Core Domain Objects:**
  - `Fruit.java` - Represents fruit items with properties like name, type, source country
  - `FruitStock.java` - Encapsulates fruit inventory data
  - `User.java` - User account information and authentication details
  - `Branch.java`, `Shop.java` - Represent bakery locations
  - `City.java`, `Country.java` - Geographical location hierarchy
- **Business Process Objects:**
  - `BorrowingRequest.java`, `BorrowingRecord.java` - Handle fruit borrowing workflow

    - `Reservation.java`, `ReservationItem.java` - Manage reservation processing
    - `ReservationSummary.java` - Aggregate reservation data
    - `Delivery.java`, `DeliveryItem.java`, `DeliveryTracking.java` - Delivery process objects
    - `DeliveryHistory.java` - Historical delivery tracking
- **Analytics and Reporting Objects:**
    - `FruitConsumption.java`, `DetailedConsumption.java` - Track usage patterns
    - `MonthlyTrend.java` - Time-based trends for forecasting
    - `SeasonConsumption.java` - Seasonal analysis of consumption
    - `CountryNeeds.java` - Aggregated country-level requirements

These beans provide a robust data model with proper encapsulation, facilitating data transfer between layers while maintaining clean separation of concerns.

1.2 Data Access Objects (`ict.db` package)

This layer handles database interaction, separating database operations from business logic:

- **Core Database Infrastructure:**
    - `DBConnection.java` - Manages database connections and configuration
    - `DBTest.java` - Testing utilities for database operations
- **Entity-Specific Data Access:**
    - `FruitDB.java` - CRUD operations for fruit types
    - `InventoryDB.java` - Stock level operations
    - `BranchFruitDB.java`, `BranchDB.java` - Branch-specific data access
    - `UserDB.java`, `UserActivityDB.java` - User management and activity tracking
    - `ShopDB.java` - Shop-specific data operations
- **Transaction Process DAOs:**
    - `BorrowingDB.java` - Borrowing request and record management
    - `ReservationDB.java`, `ReservationDB2.java` - Reservation data operations

    - `DeliveryDB.java`, `DeliveryTrackingDB.java` - Delivery management
- **Analytics and Reporting DAOs:**
    - `AnalyticsDB.java` - General analytics queries
    - `ConsumptionDB.java` - Consumption data retrieval
    - `ForecastDB.java` - Predictive analysis data
    - `StockHistoryDB.java` - Historical inventory tracking
    - `CountryNeedsDB.java` - Country-level requirement operations

This layered DAO approach abstracts database operations, allowing the rest of the application to interact with data without knowledge of the underlying database structure.

2. View Layer (User Interface)

The View component handles the presentation layer, responsible for rendering the UI and collecting user input. The view layer is organized by user role and functionality:

2.1 JSP Pages by Role

- **Common Access Pages:**
    - `login.jsp` - Authentication entry point
    - `register.jsp` - New user registration
    - `index.jsp` - Application home page
    - `redirect.jsp` - Navigation utility
- **Bakery Staff Views (`bakery` directory):**
    - `dashboard.jsp` - Main bakery staff interface
    - `fruits-stock.jsp` - Shop inventory management
    - `borrow-fruits.jsp`, `borrow-records.jsp` - Borrowing functionality
    - `reserve-fruits.jsp`, `reserve-records.jsp` - Reservation management
    - `delivery-tracking.jsp` - Track incoming deliveries
    - `profile.jsp` - User profile management
    - `view-request.jsp`, `view-reservation.jsp` - Detailed views
- **Management Views (`management` directory):**
    - `dashboard.jsp` - Executive overview

    - `analytics.jsp` - System-wide analytics
    - `profile.jsp` - Management user profile
    - **Fruit Management (`fruits` subdirectory):**
        - `index.jsp` - List all fruits
        - `create.jsp`, `edit.jsp` - Fruit type management
    - **Reporting (`reports` subdirectory):**
        - `consumption.jsp` - Consumption analytics
        - `forecast.jsp` - Predictive reports
        - `reservations.jsp` - Reservation trends
    - **User Management (`users` subdirectory):**
        - `index.jsp` - User listing
        - `create.jsp`, `edit.jsp` - User account management
- **Warehouse Staff Views (`warehouse` directory):**
    - `dashboard.jsp` - Warehouse operations overview
    - `central-warehouse.jsp` - Central warehouse inventory
    - `country-needs.jsp` - Country requirement aggregation
    - `process-reservations.jsp` - Reservation approval workflow
    - `arrange-deliveries.jsp` - Delivery coordination
    - `update-stock.jsp` - Warehouse stock management
    - `delivery-tracking.jsp` - Track outgoing deliveries
    - `profile.jsp` - Warehouse staff profile
    - `view-inventory.jsp` - Detailed inventory view

2.2 Custom Tag Implementation

The view layer is enhanced with custom tags to promote reusability and consistent presentation:

- **Tag Files (`tags` directory):**
    - `statusBadge.tag` - Visual indicator for stock status
    - `reservationStatusBadge.tag` - Reservation status visualization
- **Tag Library Descriptors (`tlds` directory):**
    - `custom.tld` - Custom tag library configuration
- **Tag Handlers (`ict.tag` package):**
    - `StatusBadgeTag.java` - Stock status badge implementation
    - `ReservationStatusBadgeTag.java` - Reservation status badge
    - `StatusColorTag.java` - Color-coding for statuses
    - `DateFormatTag.java` - Date formatting utility

2.3 Static Resources (`assets` directory)

- **CSS (`css` subdirectory)** - Stylesheets for visual presentation
- **JavaScript (`js` subdirectory)** - Client-side functionality
- **Images (`image` subdirectory)** - Visual elements

3. Controller Layer (Request Processing)

The Controller component handles HTTP requests, processes user input, and coordinates between Model and View. The controller layer is organized into functional packages:

3.1 Main Servlets (`ict.servlet` package)

- **Authentication and User Management:**
  - `LoginServlet.java` - User authentication
  - `LogoutServlet.java` - Session termination
  - `RegisterServlet.java` - New user registration
  - `ChangePasswordServlet.java` - Password management
  - `UpdateProfileServlet.java` - Profile updates
  - `UserController.java` - User CRUD operations
  - `UserActivitiesServlet.java` - Activity tracking
- **Core Business Controllers:**
  - `FruitController.java` - Fruit type management
  - `BorrowServlet.java`, `BorrowSubmitServlet.java` - Borrowing workflow
  - `ReservationRecordServlet.java` - Reservation management
  - `SelectFruitsServlet.java` - Fruit selection process
  - `DeliveryTrackingServlet.java` - Delivery monitoring
- **Analytical Controllers:**
  - `ConsumptionReportController.java` - Consumption analysis
  - `ForecastController.java` - Predictive analytics

3.2 Specialized Controller Packages

- **Inventory Management (`ict.servlet.inventory` package):**

    - `GetInventoryServlet.java` - Retrieve inventory data
    - `UpdateStockServlet.java` - Modify stock levels
    - `GetStockHistoryServlet.java` - Historical stock data
    - `WarehouseStockServlet.java` - Warehouse inventory
    - `InventoryRedirectServlet.java` - Navigation utility
- **Reservation Processing (`ict.servlet.reservation` package):**
    - `CreateReservationServlet.java` - Create reservations
    - `FixedCreateReservationServlet.java` - Enhanced reservation creation
    - `GetAvailableFruitsServlet.java` - Check availability
    - `GetWarehousesServlet.java` - Warehouse data retrieval
- **Warehouse Operations (`ict.servlet.warehouse` package):**
    - `CountryNeedsServlet.java` - Country requirement processing
    - `DeliveryManagementServlet.java` - Manage deliveries
    - `GetReservationDetailsServlet.java` - Detailed reservation info
    - `ProcessReservationsServlet.java` - Reservation workflow

3.3 Utility Classes (`ict.util` package)

- `FormatUtil.java` - Formatting utilities
- `PasswordHasher.java` - Security utility for password hashing
- `Validation.java` - Input validation services

4. MVC Interaction Flow

The implementation demonstrates a classic MVC workflow:

1. **Request Initiation:**
    - User interacts with JSP pages (View)
    - Requests are sent to appropriate Servlets (Controller)
2. **Controller Processing:**
    - Servlets validate input and extract parameters
    - Business logic is delegated to Model components
    - Controllers determine the appropriate View for response
3. **Model Operations:**
    - JavaBeans encapsulate business data and logic
    - DAO classes handle data persistence operations

    - Business rules are enforced at the Model level
4. **View Resolution:**
    - Controllers forward processing results to JSP pages
    - JSP pages use JavaBeans and custom tags to render responses
    - View components remain focused solely on presentation concerns
5. **Data Flow Example - Fruit Reservation Process:**
    - User selects fruits in `reserve-fruits.jsp` (View)
    - Form submission goes to `CreateReservationServlet.java` (Controller)
    - Controller validates input and creates `Reservation` objects (Model)
    - `ReservationDB.java` persists the reservation (Model-DAO)
    - Controller forwards to `reserve-records.jsp` or `reservation-success.jsp` (View)

5. Key MVC Implementation Strengths

1. **Clear Separation of Concerns:**
    - Model components focus exclusively on data and business logic
    - View components handle only presentation aspects
    - Controllers coordinate workflow without mixing concerns
2. **Modular Design:**
    - Functionality is organized into cohesive packages
    - Related components are grouped together (e.g., reservation-related servlets)
    - New features can be added with minimal impact on existing code
3. **Consistent Naming Conventions:**
    - Naming clearly identifies component responsibilities
    - Class names reflect their role in the MVC pattern
    - Package structure reinforces architectural boundaries
4. **Enhanced View Layer:**
    - Custom tags provide reusable UI components
    - Role-based views ensure appropriate access control
    - JSP organization matches functional requirements
5. **Robust Data Access Layer:**
    - DAO pattern abstracts database operations
    - Central database connection management

    - Entity-specific DAO classes provide focused data operations

This detailed MVC implementation enables the AIB Fruits Management System to achieve a high degree of maintainability, extensibility, and separation of concerns, facilitating both current functionality and future enhancements.

Database Structure of AIB Fruits Management System

1. Database Schema Overview

The AIB Fruits Management System is built on a relational database designed to efficiently support operations for bakery shops, warehouses, and management teams. The database structure implements a comprehensive data model that separates concerns among different functional areas while maintaining strong relational integrity between interconnected components.

2. Key Entities and Relationships

Core Entities

1. **Fruits** (`fruits` table): Central to the system, this entity stores information about different fruits including name, category, description, source country, and availability status. Each fruit has a unique identifier and belongs to a source country.
2. **Branches** (`branches` table): Represents physical locations in the AIB network, categorized as either bakery shops or warehouses. Branches are tied to specific cities and countries in a geographical hierarchy.
3. **Users** (`users` table): Stores system user information, supporting three distinct roles: bakery staff, warehouse staff, and senior management. Includes authentication data, contact information, and account status.

Inventory Management

4. **Branch Fruits** (`branch_fruits` table): A junction entity that tracks the current inventory of each fruit at each branch. It maintains important inventory metrics like:
   - Current quantity
   - Minimum threshold for reordering
   - Last update timestamp
5. **Stock History** (`stock_history` table): Records all inventory transactions with complete audit details including:
   - The user who performed the action
   - Precise quantities before and after the change
   - Type of transaction (addition, removal, reservation, checkout)
   - Timestamp and explanatory notes

Transaction Systems

6. **Borrowing System**: Consists of:
   - **Borrowing Requests** (`borrowing_requests` table): Tracks requests for borrowing fruits between branches
   - **Borrowing Items** (`borrowing_items` table): Details of individual fruits within each borrowing request
7. **Reservation System**: Manages reservations from bakeries to warehouses:
   - **Fruit Reservations** (`fruit_reservations` table): Master records for reservation requests
   - **Reservation Items** (`reservation_items` table): Individual fruits included in each reservation
8. **Delivery Tracking System**: Monitors the movement of fruits:
   - **Delivery Tracking** (`delivery_tracking` table): Master records of all deliveries
   - **Delivery Items** (`delivery_items` table): Details of fruits in each delivery
   - **Delivery History** (`delivery_history` table): Chronological timeline of status changes and events

Location Hierarchy

9. **Geographical Structure**:
   - **Countries** (`countries` table): Top level of the geographical hierarchy
   - **Cities** (`cities` table): Mid-level geographical divisions within countries
   - **Branches** (`branches` table): Physical locations within cities

Analytics Support

10. **Analytical Entities**:
    - **Consumption Analytics** (`consumption_analytics` table): Records consumption patterns by time period, location, and fruit
    - **Seasons** (`seasons` table): Defines seasonal periods for trend analysis
    - **Delivery Time Analytics** (`delivery_time_analytics` table): Aggregates delivery performance metrics

3. Functional Design Aspects

Inventory Management Support

The database structure is optimized for precise inventory tracking:

- **Real-time Stock Levels**: The branch_fruits table maintains current stock levels for each fruit at each location.
- **Threshold Alerts**: Minimum threshold values in the branch_fruits table enable automatic alerting when stock falls below critical levels.
- **Comprehensive History**: Every stock change is documented in the stock_history table, providing a complete audit trail of all inventory movements.

Inter-Branch Borrowing

The borrowing system facilitates fruit sharing between bakery shops:

- **Request Workflow**: Supports the full borrowing lifecycle (pending → approved → borrowed → returned/cancelled/rejected).
- **Item-Level Tracking**: Individual borrowed items can be tracked separately.
- **Integrated Delivery**: Borrowing requests automatically generate corresponding delivery records upon approval.

Reservation Management

The reservation system handles ordering from warehouses:

- **Priority Levels**: Supports normal, high, and urgent priority classifications for reservations.
- **Status Tracking**: Comprehensive status tracking from creation through processing to delivery.
- **Warehouse Integration**: Direct connection to warehouse inventory for availability checking.

Delivery Tracking

The delivery system monitors all fruit movements:

- **Unified Tracking**: Handles both borrowing-based and reservation-based deliveries.
- **Status Pipeline**: Manages the complete delivery lifecycle (pending → processing → in transit → out for delivery → delivered).
- **Location Awareness**: Tracks current location and estimated delivery times.
- **Event History**: Maintains a chronological history of all delivery status changes and significant events.

Analytics Capabilities

The database supports sophisticated analytical functions:

- **Consumption Patterns**: The `consumption_analytics` table records detailed consumption data by location, time period, and season.

- **Seasonal Analysis**: The `seasons` table enables seasonal trend analysis.
- **Delivery Performance**: The `delivery_time_analytics` table supports performance monitoring of delivery operations.
- **Aggregation Views**: Pre-configured database views (`country_consumption_summary`, `reservation_needs_summary`, `warehouses`) provide efficient access to commonly needed analytical summaries.

4. Design Principles Applied

1. Normalized Structure

The database implements proper normalization principles:

- **First Normal Form (1NF)**: All tables have primary keys and contain atomic values.
- **Second Normal Form (2NF)**: Non-key attributes are fully functionally dependent on the primary key.
- **Third Normal Form (3NF)**: Non-key attributes are not transitively dependent on the primary key.

For example, fruit details are stored once in the fruits table rather than being duplicated across inventory records, and junction tables like branch_fruits properly handle many-to-many relationships.

2. Referential Integrity

Strong referential integrity is maintained throughout the database:

- Foreign key constraints ensure that relationships between tables remain valid.
- Cascading actions (e.g., DELETE CASCADE) are implemented where appropriate to maintain data consistency.
- Child records cannot exist without their parent records.

3. Audit Trail Mechanisms

Comprehensive audit mechanisms track all significant system activities:

- All inventory changes are recorded with before and after values.
- User activities are logged with timestamp and IP address information.
- Delivery status transitions are documented with location and descriptive information.

4. Status-Based Workflow

Status fields throughout the database support sophisticated workflow processes:

- Enumerated types enforce valid status transitions.
- Status fields drive business process flow and user interface behavior.
- Status changes trigger appropriate system actions through database triggers.

5. Performance Optimization

The database structure is optimized for performance:

- Appropriate indexing on frequently queried fields.
- Denormalization where necessary for reporting efficiency.
- Pre-configured views for common analytical queries.
- Junction tables to efficiently handle many-to-many relationships.

5. Integration with System Components

Role-Based Access Control

The database supports the three-tiered user role system:

- **Bakery Staff**: Focus on inventory management and borrowing/reservation processes.

- **Warehouse Staff**: Emphasis on inventory management, processing reservations, and coordinating deliveries.
- **Senior Management**: Access to analytical data, user management, and system-wide reporting.

Transaction Automation

Database triggers automate critical business processes:

- Reservation creation automatically initiates delivery tracking.
- Borrowing approval triggers delivery record creation.
- Delivery status changes generate appropriate history records.
- Inventory threshold crossings can trigger alerts and notifications.

Reporting and Analytics Support

The database structure enables sophisticated reporting capabilities:

- Country and location-based consumption analytics.
- Seasonal trend analysis for demand forecasting.
- Detailed inventory movement reporting.
- Delivery performance metrics by location and fruit type.

Conclusion

The database structure for the AIB Fruits Management System provides a solid foundation that successfully balances several critical requirements:

1. **Data Integrity**: Through proper normalization, constraints, and referential integrity.
2. **Process Support**: By enabling complex workflows for borrowing, reservation, and delivery processes.
3. **Analytical Capability**: Through comprehensive data capture and optimized structures for reporting.
4. **Scalability**: By implementing efficient data structures that can accommodate growth in locations, users, and transaction volume.

5. **Security**: Through role-based access control and comprehensive audit trails.

This relational database design serves as the backbone of the entire system, enabling the seamless flow of information between different components while maintaining data consistency, security, and performance.

Brief Description of the AIB Fruits Management System

Core Design Principles and Major Characteristics

The AIB Fruits Management System represents a comprehensive enterprise solution designed to digitize and optimize fruit supply chain management across international borders. The system embodies several key design principles that address the unique challenges of AIB's multinational bakery operations:

1. Role-Based Functionality Segregation

The system implements distinct interfaces tailored to the specific responsibilities of each user role:

- **Bakery Staff Interface:** Prioritizes shop-level inventory management, borrowing capabilities, and reservation creation with intuitive visual elements that communicate stock status instantly. The card-based presentation of fruits with source country tags and visual progress bars enables staff to quickly assess availability and take appropriate actions.
- **Warehouse Staff Interface:** Focuses on aggregate inventory management, reservation processing, and delivery coordination with tabular data views optimized for handling larger volumes of information. The multi-tab approach enables warehouse staff to efficiently transition between current stock, new stock additions, and historical records.
- **Senior Managment:** Delivers analytical insights through data visualizations and consolidated metrics, emphasizing consumption trends and reservation patterns rather than day-to-day operational details. The time-series charts and

activity logs provide executives with both strategic overview and operational awareness.

2. Cross-Border Supply Chain Integration

The system successfully bridges AIB's international operations through several architectural decisions:

- **Hierarchical Location Model:** The three-tier geographical structure (country → city → branch) pervades all aspects of the system, maintaining clear context for all inventory and transactions.
- **Source Country Traceability:** Fruits consistently maintain their origin information throughout all processes, with visual country tags providing immediate source identification.
- **Consolidated Needs Aggregation:** The system automatically aggregates individual shop reservation requests at the country level, enabling efficient bulk ordering from source warehouses.

3. Intuitive Visual Communication System

The interface employs a consistent visual language that reduces cognitive load and improves efficiency:

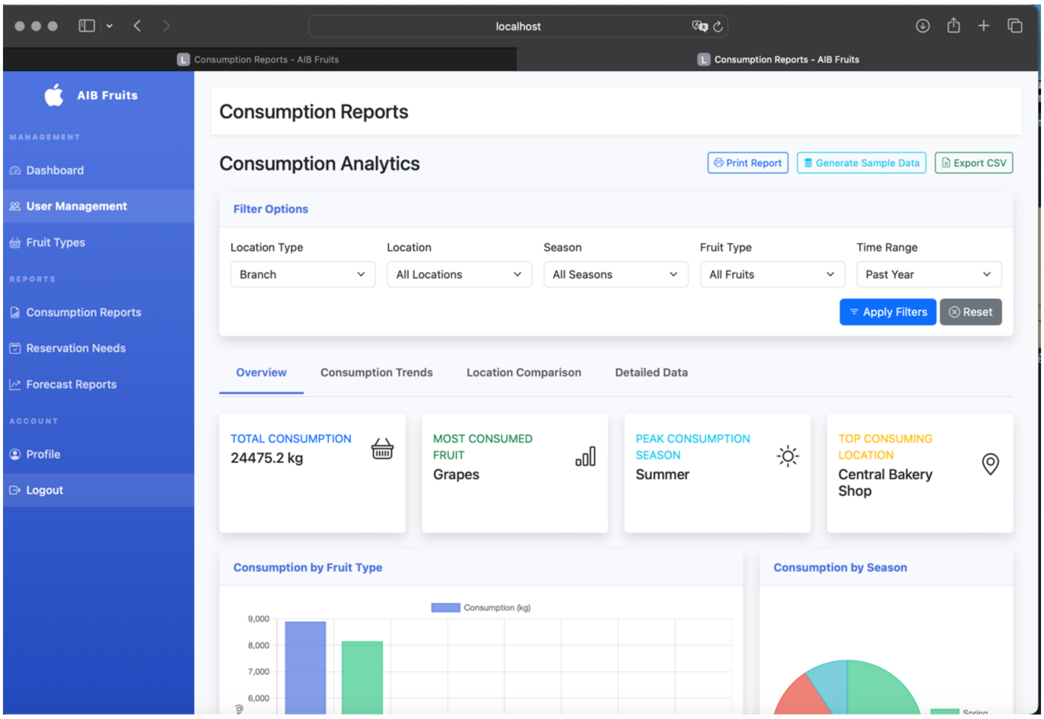
- **Status-Based Color Coding:** Green/yellow/red visual indicators throughout the system provide instant status recognition based on predefined thresholds.
- **Progressive Disclosure Pattern:** Complex information is revealed progressively, with summary views expanding to detailed information as needed.
- **Contextual Guidance:** Information boxes provide explanatory content at appropriate points in workflows, enhancing usability without requiring extensive training.
- **Consistent Action Patterns:** Similar actions (update, history, filter) maintain consistent positioning and styling across different sections, reducing learning curve.

4. Strategic Layout Design

The system implements a consistent and thoughtful layout structure that enhances usability across all interfaces:

- **Persistent Navigation Sidebar:** A fixed left sidebar provides role-appropriate navigation options using consistent iconography and clear categorization (e.g., "Warehouse Operations," "Inventory Management," "Account"). This ensures users always have access to core functions regardless of their current task.
- **Context-Aware Header Bar:** The header maintains user awareness of their location, role, and identity while providing consistent branding. The green header for warehouse staff versus blue for management creates immediate visual differentiation of interfaces.
- **Card-Based Information Grouping:** Summary metrics and status information are presented in clean, contained card elements with ample white space, making complex information easily scannable. This is particularly evident in the management dashboard's KPI cards and the bakery staff's fruit inventory cards.
- **Hierarchical Information Presentation:** Information is organized from general to specific, with high-level metrics at the top, filtering options in the middle, and detailed data below. This creates a natural information flow that guides users from overview to specifics.
- **Tab-Based Workflow Organization:** Related functions are grouped into intuitive tabs (e.g., "Current Stock," "Add New Stock," "Stock History"), allowing users to switch between related tasks while maintaining contextual awareness of their current process area.

Example of the Senior Management page :



Consumption Reports

Consumption Analytics

Filter Options

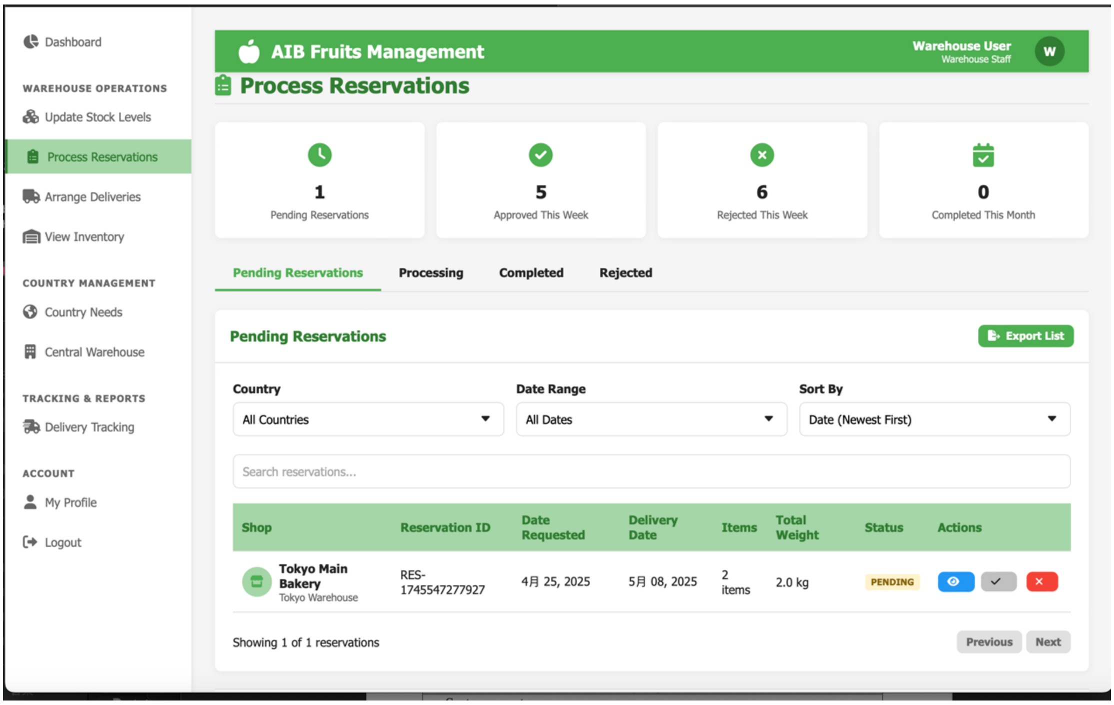
Location Type	Location	Season	Fruit Type	Time Range
Branch	All Locations	All Seasons	All Fruits	Past Year

TOTAL CONSUMPTION	MOST CONSUMED FRUIT	PEAK CONSUMPTION SEASON	TOP CONSUMING LOCATION
24475.2 kg	Grapes	Summer	Central Bakery Shop

Consumption by Fruit Type

Consumption by Season

Example of the Warehouse Staff page :



Process Reservations

1	5	6	0
Pending Reservations	Approved This Week	Rejected This Week	Completed This Month

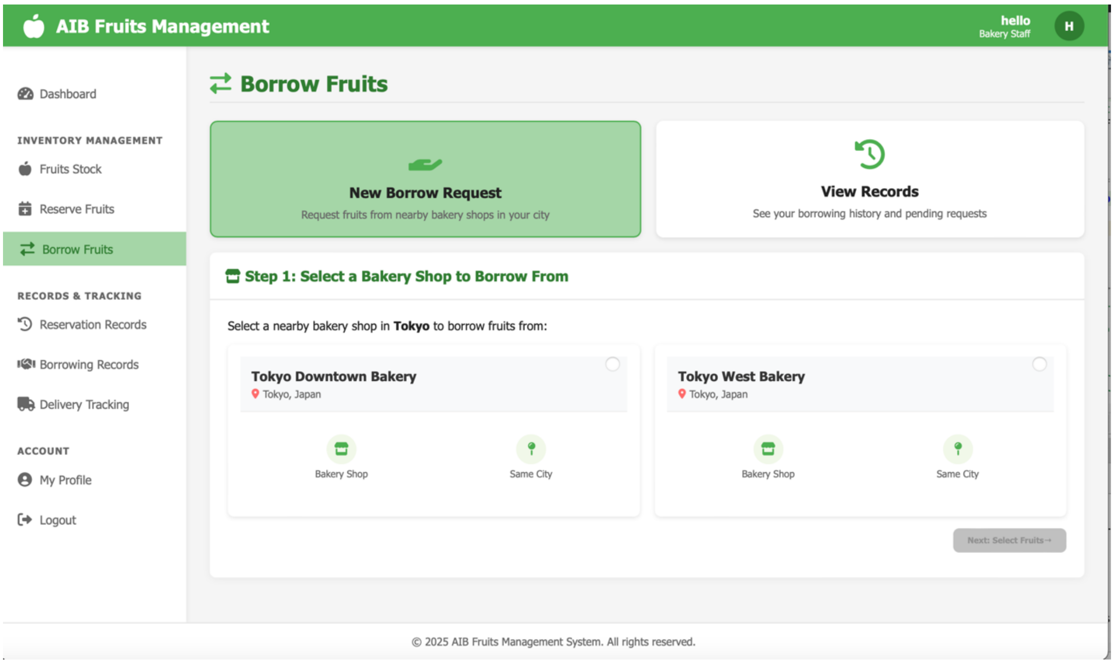
Pending Reservations

Country	Date Range	Sort By
All Countries	All Dates	Date (Newest First)

Shop	Reservation ID	Date Requested	Delivery Date	Items	Total Weight	Status	Actions
Tokyo Main Bakery Tokyo Warehouse	RES-1745547277927	4月 25, 2025	5月 08, 2025	2 items	2.0 kg	PENDING	👁 ✓ ✕

Showing 1 of 1 reservations

Example of the Bakery Staff:



Request fruits from nearby bakery shops in your city

See your borrowing history and pending requests

Step 1: Select a Bakery Shop to Borrow From

Select a nearby bakery shop in **Tokyo** to borrow fruits from:

Tokyo Downtown Bakery
Tokyo, Japan

Tokyo West Bakery
Tokyo, Japan

© 2025 AIB Fruits Management System. All rights reserved.

5. Process-Oriented Workflow Design

Rather than focusing purely on data management, the system structures user interactions around complete business processes:

- **End-to-End Borrowing Process:** The system guides users through the complete borrowing lifecycle from request creation through approval to return.
- **Reservation-to-Delivery Pipeline:** Reservation requests automatically flow into delivery tracking, creating a seamless transition between related processes.
- **Tab-Based Process Segmentation:** Related activities are grouped into logical tabs, allowing users to focus on specific process stages without distraction.
- **Integrated Historical Context:** Each process area provides access to historical records, enabling users to reference past activities within the same functional context.

6. Data-Driven Decision Support

The system transforms operational data into actionable insights through several mechanisms:

- **Real-Time Stock Visualization:** Current inventory levels are consistently presented alongside minimum thresholds, enabling proactive inventory management.
- **Consumption Trend Analysis:** The management dashboard presents time-series visualizations of consumption patterns by fruit type, enabling trend identification.
- **Activity Monitoring:** Chronological activity feeds provide transparency into system events, user actions, and process completions.
- **Status Summarization:** Key metrics relevant to each user role are highlighted prominently, from stock weights for warehouse staff to pending reservation counts for management.

Technical Implementation Highlights

From a technical perspective, the AIB Fruits Management System demonstrates sophisticated implementation of Java EE technologies:

- **Clean MVC Separation:** The system maintains strict separation between presentation (JSP views), business logic (JavaBean models), and request handling (Servlet controllers).
- **Custom Tag Integration:** Custom JSP tags enhance the view layer with reusable components for status badges and date formatting, promoting UI consistency.
- **Layered DAO Architecture:** The data access layer abstracts database operations through entity-specific DAO classes, providing clean separation from business logic.
- **Responsive Design Implementation:** The interface fluidly adapts to different screen sizes while maintaining usability and information hierarchy.

The AIB Fruits Management System successfully balances the complexity of international supply chain management with an intuitive, role-appropriate user experience. Through thoughtful design decisions and robust technical implementation, the system transforms AIB's fruit inventory management from a manual, error-prone process into a streamlined, data-driven operation that supports the company's multinational bakery business.

Conclusion

The AIB Fruits Management System represents a successful implementation of enterprise systems concepts to address complex supply chain challenges in a multinational retail environment. Through careful analysis of business requirements and thoughtful application of Java EE technologies, the project has delivered a comprehensive solution that transforms AIB's fruit inventory management processes.

Key Achievements

The system has successfully addressed all the primary challenges identified in AIB's original manual processes:

1. **Enhanced Inventory Visibility:** By implementing real-time tracking of fruit stock levels across all locations, the system has eliminated the information gaps that previously led to supply shortages and excess inventory. The visual status indicators and intuitive interfaces provide immediate awareness of inventory status, enabling proactive management.
2. **Streamlined Cross-Border Operations:** The hierarchical geographical model and consistent source country tracking have created a unified view of AIB's international supply chain. This integration enables seamless coordination between source country warehouses, central warehouses, and local bakery shops across different countries.
3. **Efficient Inter-Shop Resource Sharing:** The borrowing functionality has transformed how shops within the same city can share resources, reducing waste and minimizing emergency shortages. This peer-to-peer sharing capability creates a more resilient local network of shops that can support each other's immediate needs.
4. **Structured Reservation Process:** The 14-day advance reservation system has introduced predictability into AIB's international fruit supply chain. By aggregating reservation requests at the country level, the system enables more efficient ordering from source countries and optimizes logistics planning.
5. **Data-Driven Decision Support:** The analytical capabilities provided to senior management have transformed operational data into strategic insights. Consumption trends, reservation patterns, and inventory metrics now support informed decision-making about seasonal planning, regional needs, and resource allocation.

Business Impact

The implementation of the AIB Fruits Management System is expected to deliver significant business benefits:

- **Reduced Operational Costs:** More efficient inventory management and optimized ordering will minimize both wastage and emergency sourcing costs.
- **Improved Product Quality:** Better planning and coordination ensure fresher fruits are available, maintaining AIB's commitment to quality ingredients.
- **Enhanced Staff Productivity:** Role-specific interfaces and streamlined workflows allow staff to focus on their core responsibilities rather than administrative coordination.
- **Greater Supply Chain Resilience:** The integrated view of inventory across locations creates better responsiveness to supply disruptions or unexpected demand fluctuations.
- **Strategic Growth Support:** The system's scalable architecture will accommodate AIB's future expansion to new international locations.

Technical Excellence

The project demonstrates technical excellence through several key aspects:

1. **Clean Architecture:** The strict adherence to MVC principles creates a maintainable and extensible codebase that separates concerns appropriately.
2. **User-Centered Design:** The role-based interfaces with consistent visual language prioritize usability and efficiency for each user group.
3. **Database Optimization:** The normalized database structure with appropriate relationships ensures data integrity while supporting complex analytical queries.
4. **Security Implementation:** Role-based access control, secure authentication, and proper session management protect sensitive business data.
5. **International Support:** Multilingual elements and cultural considerations in the interface accommodate AIB's diverse international operations.

Future Opportunities

While the current implementation fulfills all core requirements, several opportunities for future enhancement could further extend the system's capabilities:

1. **Predictive Analytics:** Implementing AI-powered demand forecasting based on historical consumption data could further optimize ordering and inventory management.
2. **Mobile Applications:** Developing dedicated mobile apps for bakery and warehouse staff could enhance operational flexibility and efficiency.
3. **Supplier Integration:** Extending the system to provide limited access to fruit suppliers could streamline communication and ordering processes.
4. **IoT Integration:** Incorporating sensors for temperature monitoring during fruit transportation could enhance quality control.

Skills Checklist

The AIB Fruits Management System implements a wide range of Java EE technologies and web development concepts. Below is a comprehensive list of the skills and technologies applied in the project:

Checklist

Skill/Technology	Implementation Details
MVC Architecture	Applied clear separation between models (beans), views (JSP pages), and controllers (servlets) throughout system structure
JSP/Servlets for Dynamic HTML Generation	Multiple role-specific JSP pages (Eg:bakery_inventory.jsp, management dashboards) with dynamic content generation based on database results
JSP/Servlets for Form Processing	Form submission handling in multiple servlets (Eg:LoginServlet, UpdateProfileServlet, UserController) with parameter extraction and validation
JSP Actions	Used <jsp:useBean>, <jsp:setProperty>, and <jsp:include> for component management and bean integration
Custom Tags (Taglib)	Created custom tags for status badges (statusBadge.tag, reservationStatusBadge.tag) and date formatting

	(statusBadge.java,ReservationStatusBadgeTag.java,DateFormatTag.java) with associated TLD configuration
JavaBeans	Implemented comprehensive domain objects (Eg:Fruit.java, User.java, Delivery.java) with proper encapsulation and business logic
Batch SQL Processing	Used JDBC batch operations for improved performance when inserting multiple records (Eg:BorrowingDB.createMultipleBorrowingRequests)
JDBC for Database Access	Developed DAO classes (Eg:FruitDB.java, UserDB.java) with prepared statements, connection pooling, and transaction management
Session Management	Implemented secure session handling with appropriate checks and timeouts throughout application
Login Control	Created role-based authentication system (Eg:LoginServlet.java) with password hashing and role-specific redirection
Responsive Web Design	Implemented mobile-friendly interfaces using CSS media queries and flexible layouts
RESTful API Design	Created consistent API endpoints with proper HTTP verb usage (GetStockHistoryServlet)
Interactive UI Components	Created dynamic filtering controls, modal dialogs, and real-time updates with JavaScript
Data Visualization	Implemented interactive charts for consumption trends and seasonal forecasting using Chart.js
Form Validation	Applied both client-side and server-side validation for all user inputs
AJAX for Asynchronous Updates	Used fetch API to retrieve stock history and update inventory without page reloads
JSON Processing	Implemented JSON serialization/deserialization with Gson for API responses
Error Handling	Comprehensive error detection, user-friendly messages, and exception logging
Secure Coding Practices	Applied prepared statements to prevent SQL injection, input validation, and password hashing
State Management	Implemented proper application state management across multiple workflows

Advanced Technical Implementations

The project demonstrates several advanced implementation techniques:

- **Custom Tag Lifecycle Management**: Properly implemented tag lifecycle methods with attribute handling
- **Transactional Database Operations**: Ensured data integrity through proper transaction handling
- **Hierarchical Data Modeling**: Implemented the complex geographical hierarchy (country → city → branch)
- **Stateful Process Workflows**: Created multi-step workflows for reservation and delivery processes
- **Dynamic Data Filtering**: Implemented both server-side and client-side filtering options
- **Real-time Status Visualization**: Created intuitive visual representations of inventory levels
- **Cross-browser Compatibility**: Ensured consistent behavior across different browsers
- **Code Modularity**: Developed reusable components throughout the codebase

The AIB Fruits Management System successfully implements all required Java EE technologies while incorporating modern web development practices to create an intuitive, efficient enterprise application.