



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the implementation details for **AgroTrack - Farm Produce Logistics API**. The API is a Spring Boot REST service for tracking farm produce logistics, including shipment status, inventory, and delivery scheduling. The system uses JPA for ORM, MySQL for persistence, JWT for security, and is containerized with Docker.

1. System Components

Component	Description	Key Responsibilities
REST API Layer	Spring Boot Controllers	Expose endpoints, handle requests
Service Layer	Business Logic Services	Process business logic, validation
Repository Layer	JPA Repositories	Data access, CRUD operations
Security Module	JWT-based Security	Authentication, authorization
Persistence Layer	MySQL Database	Data storage
Deployment	Docker	Containerization, environment config

2. Class/Interface Overview

Class/Interface	Description	Key Methods/Attributes
ProduceController	REST endpoints	getAll(), create(), update(), delete()
ShipmentController	Shipment endpoints	track(), schedule(), updateStatus()
InventoryController	Inventory endpoints	getStock(), updateStock()
DeliveryController	Delivery endpoints	scheduleDelivery(), getStatus()
ProduceService	Business logic	findAll(), save(), update(), delete()
ShipmentService	Shipment logic	trackShipment(), scheduleShipment()
InventoryService	Inventory logic	getInventory(), updateInventory()
DeliveryService	Delivery logic	schedule(), getDeliveryStatus()
ProduceRepository	JPA Repository	findById(), save(), delete()
User, Role	Security entities	JWT claims, roles
JwtFilter, AuthService	Security filters/services	doFilter(), authenticate()



3. Data Structure Overview

Entity	Attributes	Relationships
Produce	id, name, type, quantity, unit, createdAt	One-to-Many: Shipments
Shipment	id, produceId, status, scheduledAt, deliveredAt	Many-to-One: Produce
Inventory	id, produceId, stockLevel, lastUpdated	One-to-One: Produce
Delivery	id, shipmentId, scheduledDate, status, deliveredDate	One-to-One: Shipment
User	id, username, password, roles	Many-to-Many: Role
Role	id, name	Many-to-Many: User

4. Algorithms/Logic

Shipment Tracking Flow (Pseudocode):

```
// Track Shipment Status
Shipment trackShipment(Long shipmentId) {
    Shipment s = shipmentRepository.findById(shipmentId);
    if (s == null) throw new NotFoundException();
    return s;
}

// Schedule Delivery
Delivery scheduleDelivery(Long shipmentId, Date date) {
    Shipment s = shipmentRepository.findById(shipmentId);
    if (s.status != 'READY') throw new InvalidStateException();
    Delivery d = new Delivery(shipmentId, date, 'SCHEDULED');
    return deliveryRepository.save(d);
}
```

5. Error Handling

Scenario	Handling Approach
Entity Not Found	Return 404 Not Found
Invalid Input/State	Return 400 Bad Request
Unauthorized Access	Return 401 Unauthorized
Forbidden Operation	Return 403 Forbidden
Database/Server Error	Return 500 Internal Server Error
JWT Expired/Invalid	Return 401 Unauthorized, error message



End of Document