KUBOTA DMS - Detailed Sequence Diagrams

This document contains comprehensive sequence diagrams for the KUBOTA Dealer Management System (DMS), illustrating the main business flows and system interactions.

Table of Contents

1. User Authentication & Authorization Flow

2. Warranty Claim Request (WCR) Flow

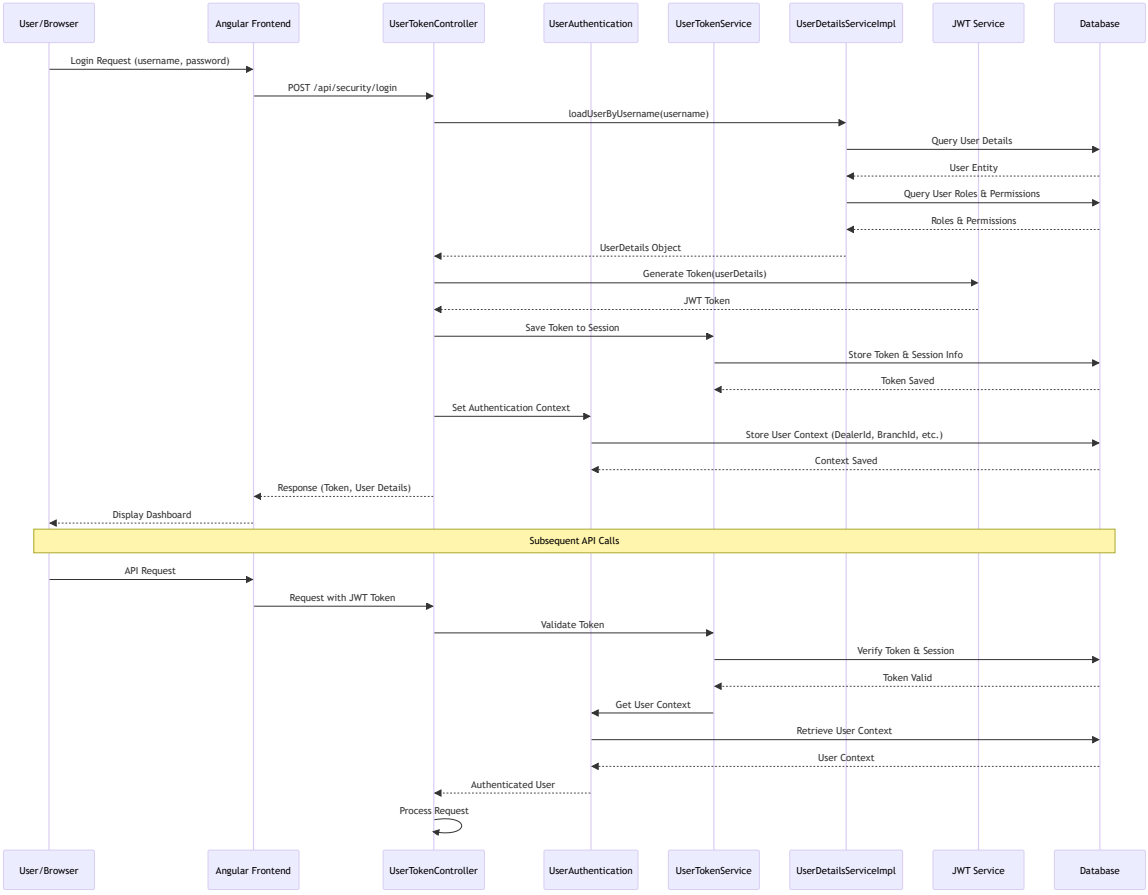
3. Spare Parts Sales Order Flow

4. Service Job Card Creation Flow

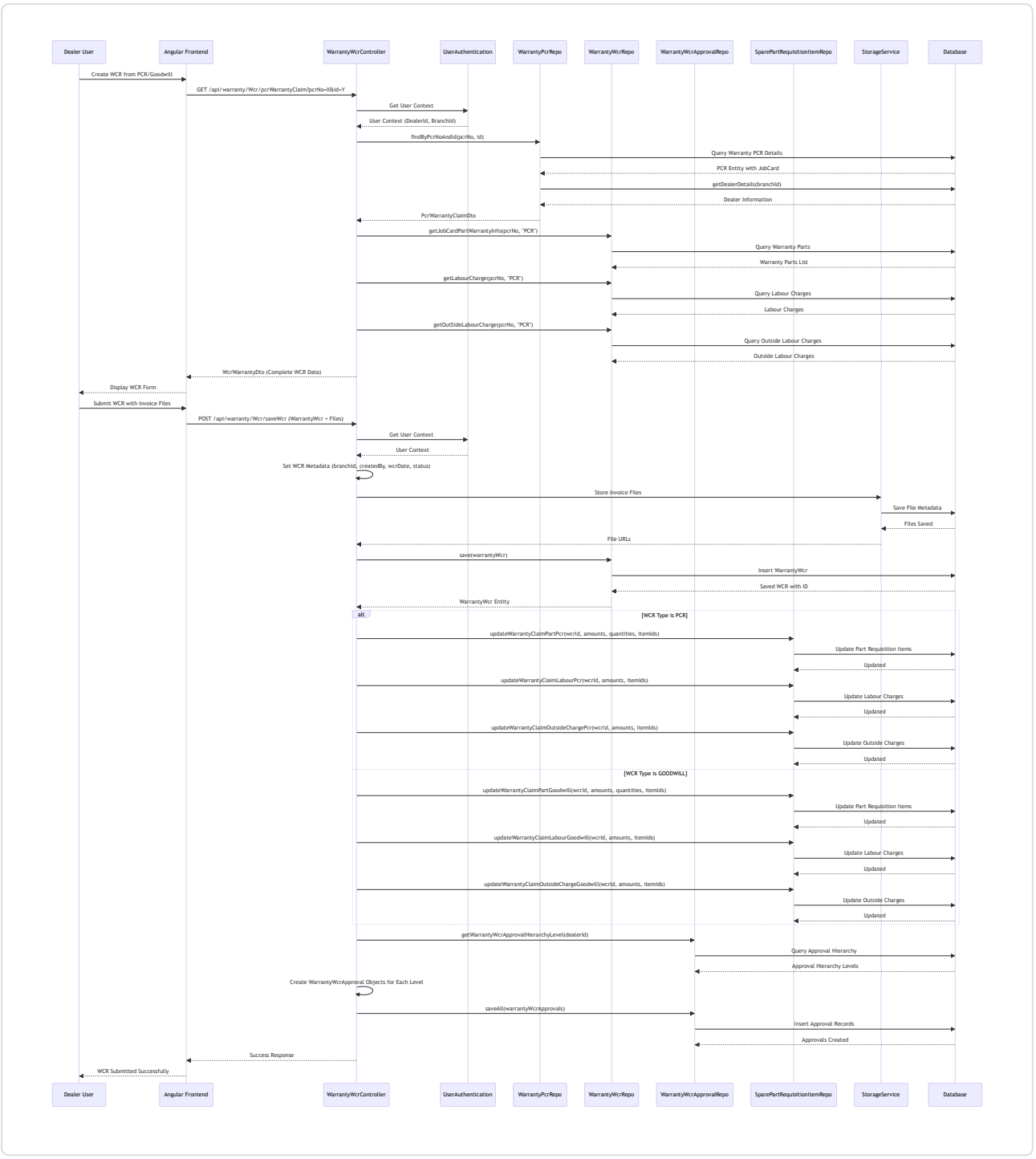
5. File Upload & Storage Flow

6. Approval Workflow Flow

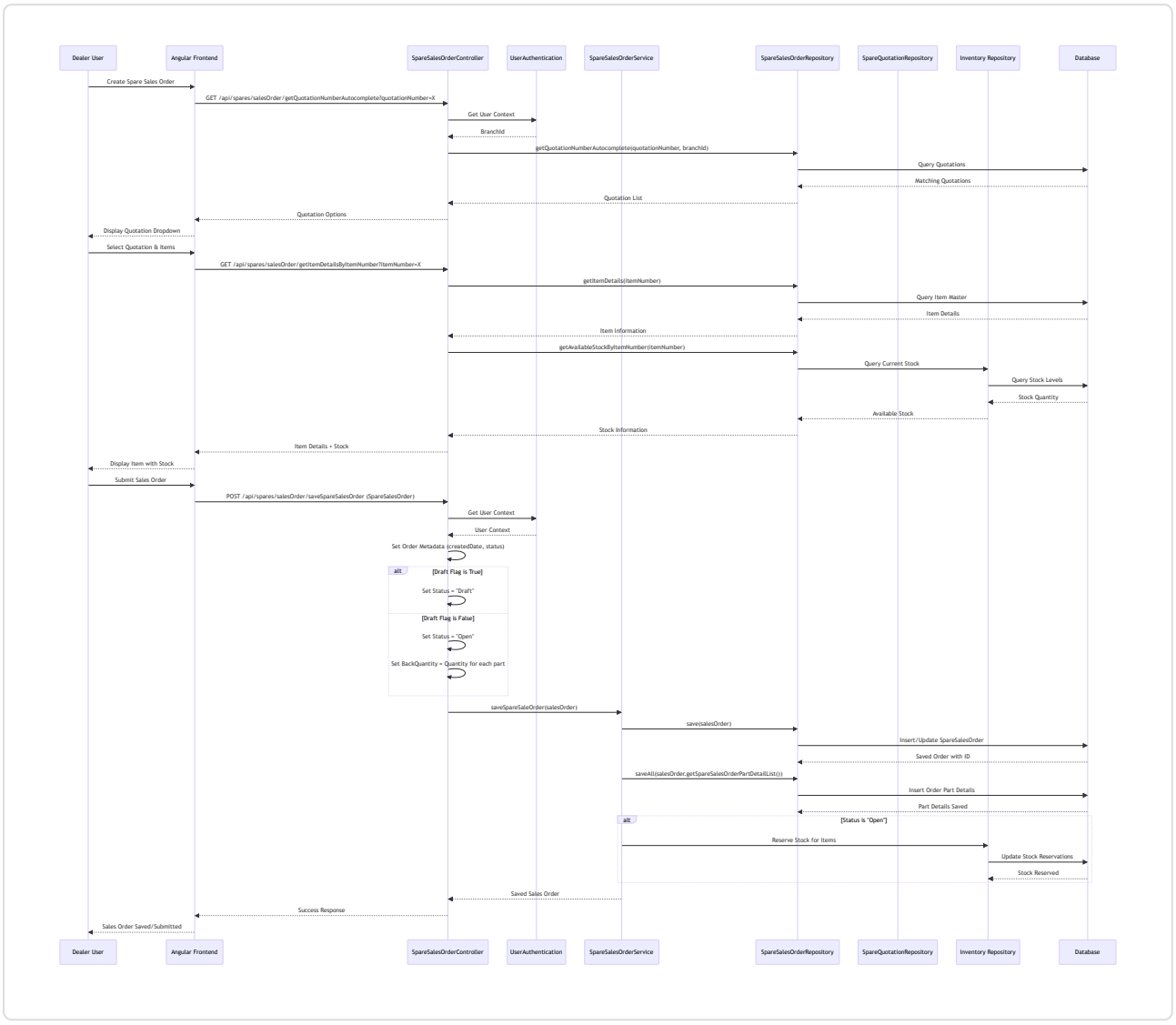
1. User Authentication & Authorization Flow



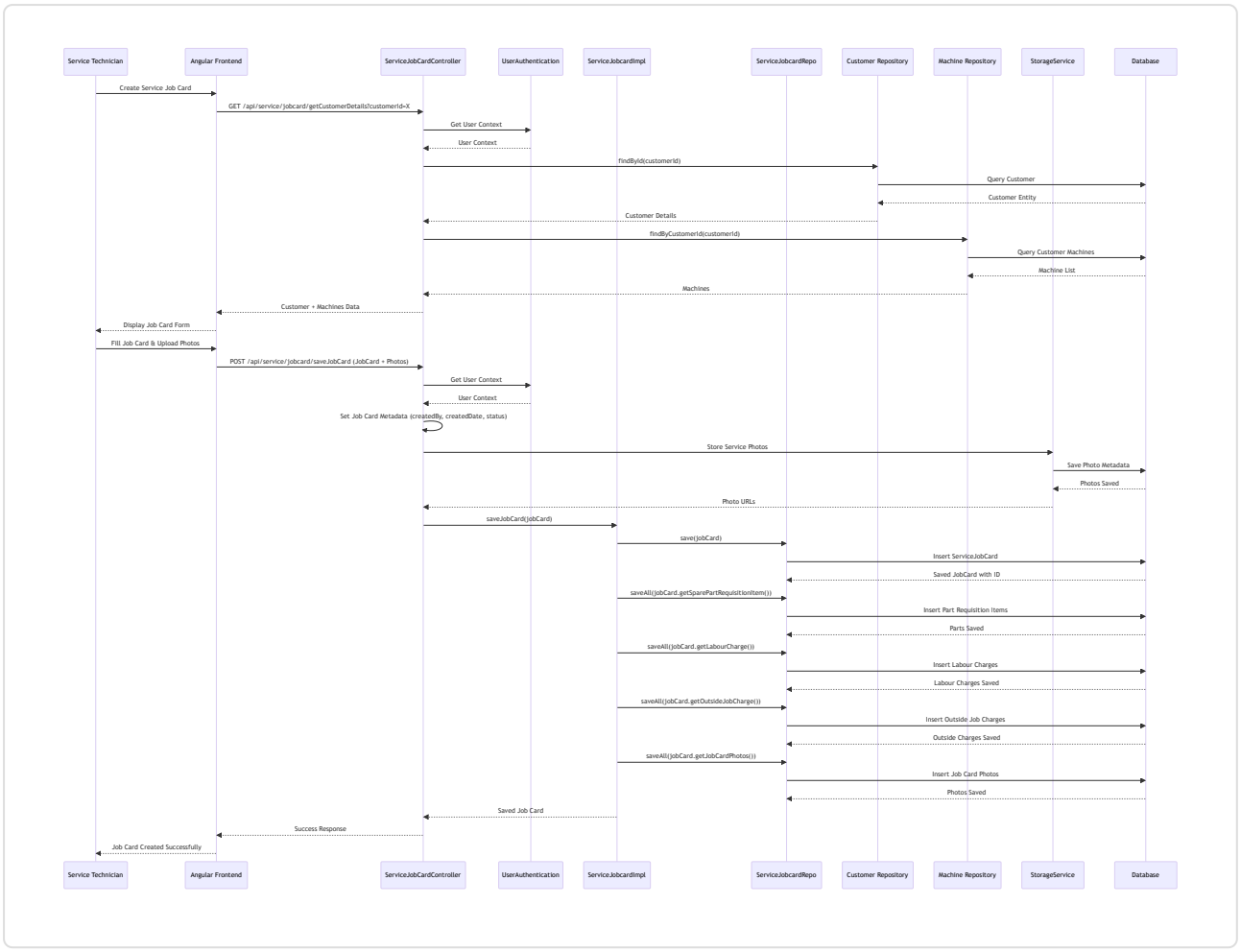
2. Warranty Claim Request (WCR) Flow



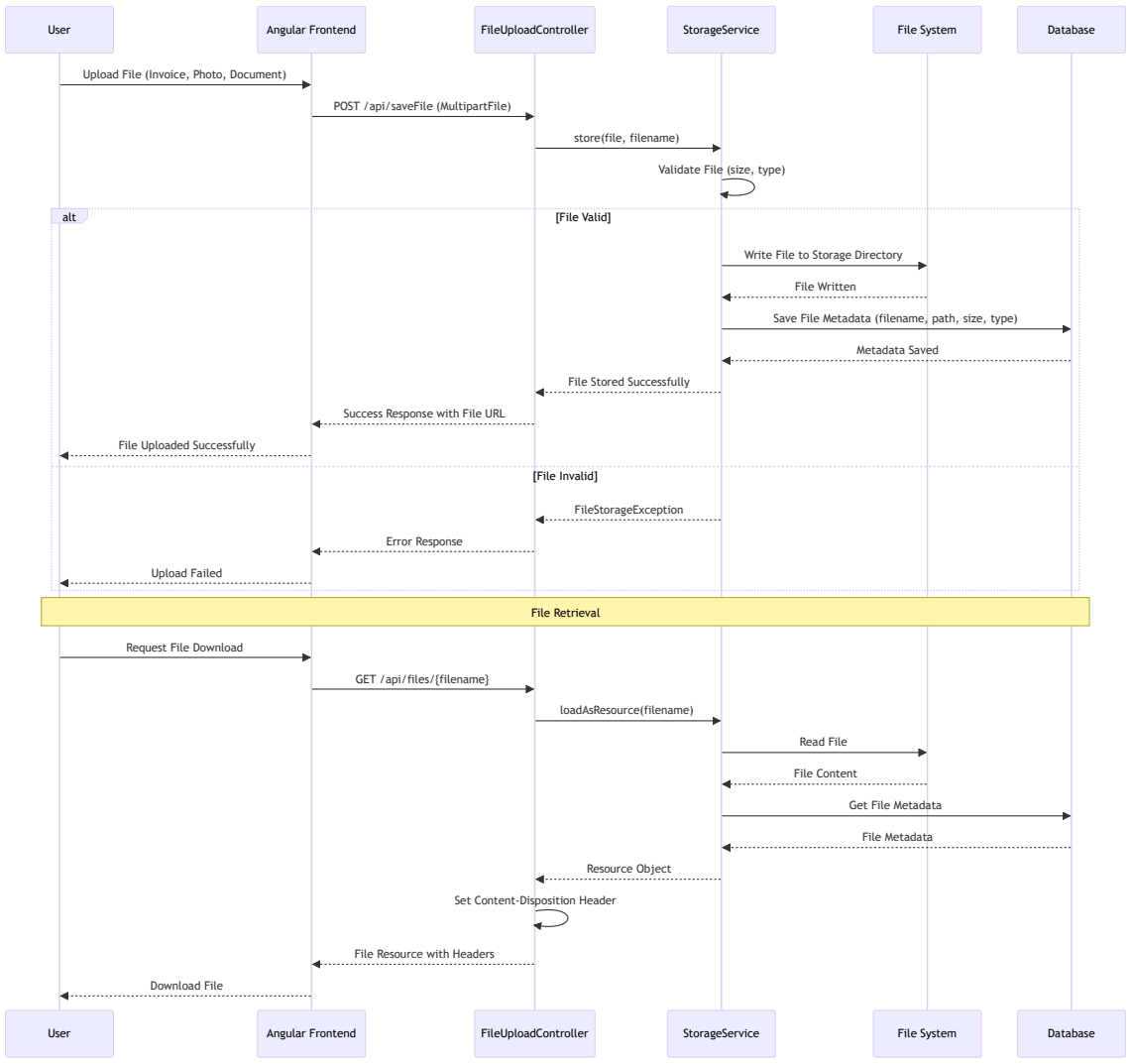
3. Spare Parts Sales Order Flow



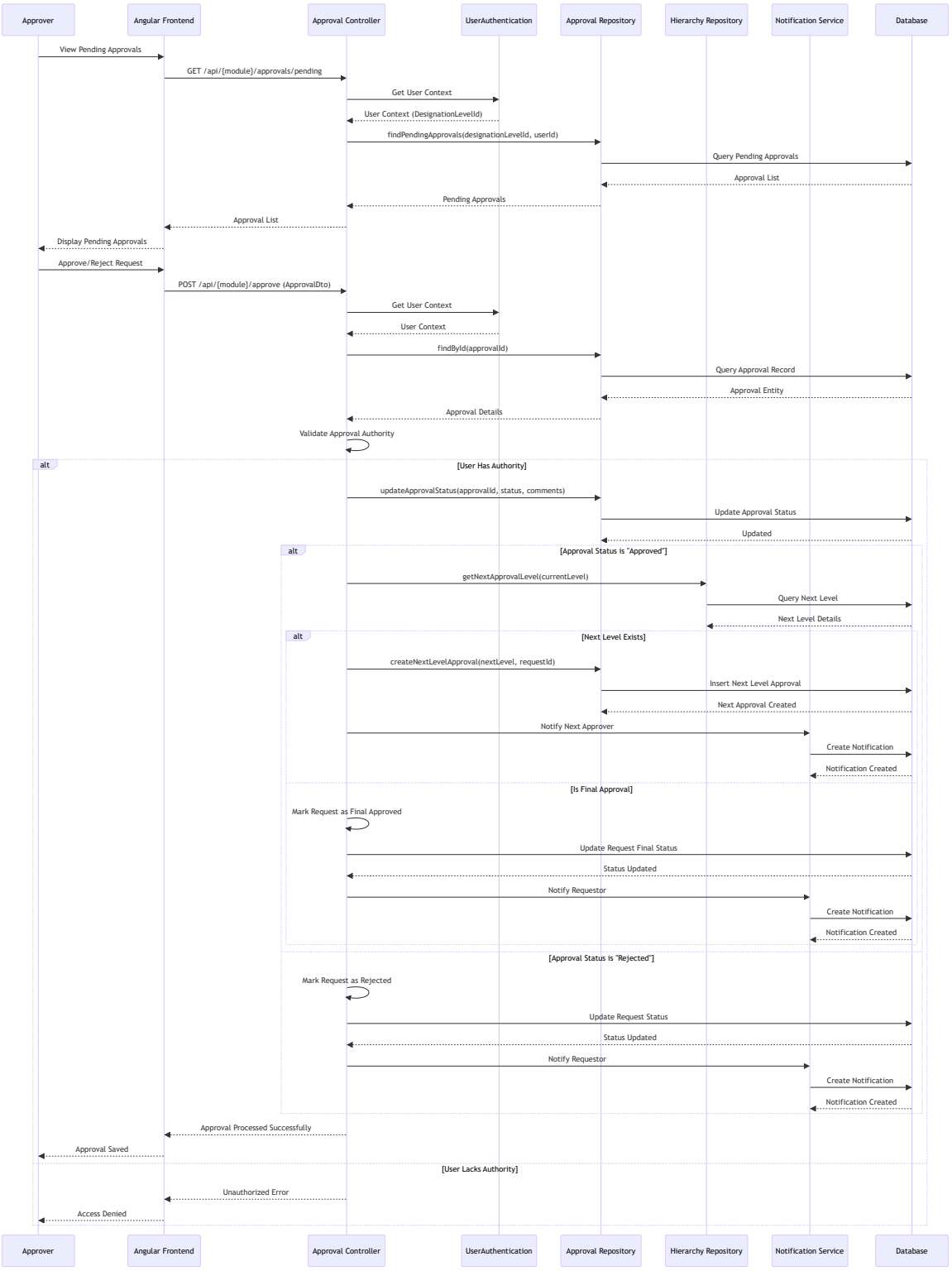
4. Service Job Card Creation Flow



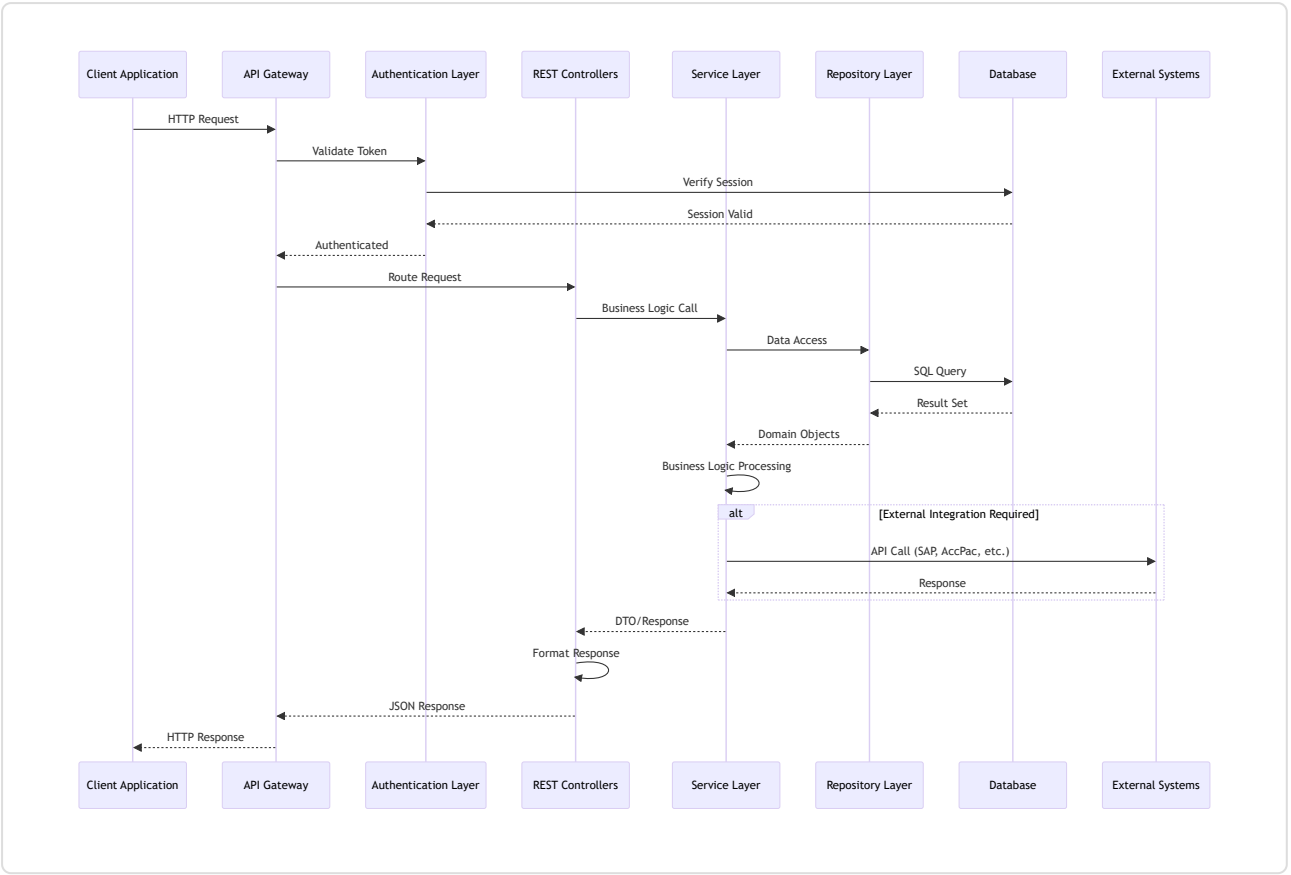
5. File Upload & Storage Flow



6. Approval Workflow Flow


Approver
Angular Frontend
Approval Controller
UserAuthentication
Approval Repository
Hierarchy Repository
Notification Service
Database
View Pending Approvals
GET /api/{module}/approvals/pending
Get User Context
User Context (DesignationLevelId)
findPendingApprovals(designationLevelId, userId)
Query Pending Approvals
Approval List
Pending Approvals
Approval List
Display Pending Approvals
Approve/Reject Request
POST /api/{module}/approve (ApprovalDto)
Get User Context
User Context
findById(approvalId)
Query Approval Record
Approval Entity
Approval Details
Validate Approval Authority
alt
[User Has Authority]
updateApprovalStatus(approvalId, status, comments)
Update Approval Status
Updated
alt
[Approval Status is "Approved"]
getNextApprovalLevel(currentLevel)
Query Next Level
Next Level Details
alt
[Next Level Exists]
createNextLevelApproval(nextLevel, requestId)
Insert Next Level Approval
Next Approval Created
Notify Next Approver
Create Notification
Notification Created
[Is Final Approval]
Mark Request as Final Approved
Update Request Final Status
Status Updated
Notify Requestor
Create Notification
Notification Created
[Approval Status is "Rejected"]
Mark Request as Rejected
Update Request Status
Status Updated
Notify Requestor
Create Notification
Notification Created
Approval Processed Successfully
Approval Saved
[User Lacks Authority]
Unauthorized Error
Access Denied
Approver
Angular Frontend
Approval Controller
UserAuthentication
Approval Repository
Hierarchy Repository
Notification Service
Database

System Architecture Overview



Notes

Key Components:

- **Controllers**: Handle HTTP requests/responses, input validation
- **Services**: Business logic, orchestration, transaction management
- **Repositories**: Data access layer, database operations
- **Authentication**: JWT-based authentication, user context management
- **Storage Service**: File upload/download management
- **Approval System**: Multi-level approval workflow

Common Patterns:

1. All requests go through authentication/authorization
2. User context (DealerId, BranchId) is injected via UserAuthentication
3. File uploads are handled by StorageService
4. Approvals follow hierarchical workflow
5. Database operations use JPA/Hibernate repositories

Error Handling:

- Global exception handler catches all exceptions
- ApiResponse wrapper for consistent response format

- Validation errors returned with appropriate HTTP status codes

---

Last Updated: December 2025