

Technical Design Document

Feature: Document Upload & View in Purchase Order Workflow

1. Objective

Enable users to upload supporting documents (e.g., quotations, specifications, approval notes) while creating a Purchase Order (PO). These documents should be accessible to approvers during the PO approval process.

2. Scope

In Scope

- Upload of multiple documents at the time of PO creation.
- Storage of documents in the backend.
- Retrieval and view of documents during PO approval.
- Role-based access control.

Out of Scope

- Post-approval document upload.
- Document versioning.
- OCR or content extraction.

3. Functional Requirements

3.1 Upload During PO Creation

- Attach one or more files.
- Supported formats: PDF, DOCX, XLSX, JPG, PNG.
- File size limit: Configurable.
- Documents linked to PO record.

3.2 View During Approval

- Approvers can view/download uploaded documents.
- Show metadata.
- Inline preview for PDFs/images.
- View-only access for approvers.

4. Non-Functional Requirements

- Security: Access control, virus scanning.
- Performance: Efficient retrieval.
- Scalability: Handle multiple documents.
- Auditability: Maintain logs of uploads, views.

5. System Design

Frontend: Upload widget in PO creation, viewer in approval screen.

Backend: APIs for upload and retrieval.

Storage: Prefer Cloud (S3/Azure/GCP).

Database Schema:

| Column | Type | Description |
|-------------|--------------|---------------------------------|
| id | BIGINT (PK) | Primary key |
| po_id | BIGINT (FK) | Reference to purchase_orders.id |
| file_name | VARCHAR(255) | Name of file |
| file_type | VARCHAR(50) | Type of file |
| file_size | BIGINT | Size of file in bytes |
| storage_url | VARCHAR(500) | Storage location (URL/Path) |

| | | |
|-------------|-------------|-------------------|
| uploaded_by | BIGINT (FK) | User who uploaded |
| uploaded_at | TIMESTAMP | Upload date/time |

6. User Interface

PO Creation: Drag & Drop/Browse file upload widget.

Approval: List of attached documents with view/download.

7. Security Considerations

- Validate file type & size.
- Secure storage.
- Signed URLs for downloads.
- Encryption at rest.
- HTTPS for all operations.

8. Implementation Plan

1. Extend DB schema.
2. Develop backend APIs.
3. Integrate file upload widget.
4. Add document viewer.
5. Apply role-based access.
6. Test security and performance.

9. Future Enhancements

- Upload during approval.
- Document versioning.
- Integration with DMS.
- Tagging and categorization.