Local Agentic Document Extraction System

Complete Project Plan

4-Agent Architecture with 3-Layer Anti-Hallucination

Specification	Value
Model	Qwen3-VL 8B
Backend	LM Studio (Local)
Framework	LangGraph
Agents	4 Specialized
Validation	3-Layer Anti-Hallucination
Compliance	HIPAA Ready
Timeline	12 Weeks
Team Size	2-3 Engineers

Table of Contents

1. Project Overview

Build a production-ready, HIPAA-compliant document extraction system using local Vision Language Models (VLM) with a 4-agent architecture for healthcare Revenue Cycle Management (RCM) documents.

Timeline Summary

Week	Phase	Focus
1	Phase 0	Prerequisites & Setup
2-3	Phase 1	Core Infrastructure
4-6	Phase 2	Agent Framework
7-8	Phase 3	Anti-Hallucination System
9-10	Phase 4	Integration & Testing
11-12	Phase 5	Deployment
13-14	Buffer	Contingency

2. System Architecture

4-Agent Pipeline

Agent	Role	VLM Calls	Key Functions
Orchestrator	State Machine	0	Workflow control, error handling, checkpointing
Analyzer	Document Understanding	1/doc	Classification, structure detection, schema selection
Extractor	Data Extraction	2/page	Schema-driven extraction, dual-pass verification
Validator	Quality Assurance	0-1/doc	Validation, hallucination detection, output formatting

Data Flow

PDF Input → Preprocessor → Orchestrator → Analyzer (1 VLM call) → Extractor (2 VLM calls/page) → Validator (0-1 VLM call) → JSON Output

4. Phase 0: Prerequisites & Setup (Week 1)

Duration: 1 Week | Resources: 1 Engineer + IT Support

Task 0.1: Hardware Procurement

Component	Minimum	Recommended
GPU	RTX 3090 (24GB)	RTX 4090 (24GB) or A6000
RAM	32GB DDR4	64GB DDR5
CPU	8-core	16-core
Storage	500GB NVMe	1TB NVMe

Task 0.2-0.3: LM Studio & Model Setup

1. Install LM Studio from lmstudio.ai
2. Enable vision model support
3. Download Qwen3-VL 8B (Q4_K_M quantization, ~6GB VRAM)
4. Configure server on port 1234
5. Set context length to 32768

Task 0.4-0.5: Python Environment

Key Dependencies: langgraph, PyMuPDF, openai, pydantic, Pillow, tenacity, fastapi, uvicorn, celery, redis, cryptography, prometheus-client, pytest

Milestone: VLM responds correctly to image + text prompt via localhost:1234

5. Phase 1: Core Infrastructure (Weeks 2-3)

Duration: 2 Weeks | Resources: 2 Engineers

Week 2: PDF Processing Pipeline

1. Build PDF ingestion module with validation
2. Implement page extraction at 300 DPI
3. Add quality enhancement (contrast, denoise, deskew)
4. Create batch processing manager
5. Write unit tests (>80% coverage)

Week 2: LM Studio Client

1. Create connection manager with health monitoring
2. Build vision request handler (base64 encoding)
3. Implement response parser with JSON extraction
4. Add retry logic with tenacity (3 attempts, exponential backoff)
5. Write unit tests

Week 3: Schema System

1. Define base schema structure with Pydantic
2. Create field validators (date, currency, NPI, ICD-10, CPT)
3. Build CMS-1500 schema with cross-field rules
4. Build EOB/Remittance schema
5. Add cross-field rule engine

Deliverables: PDF processor, LM client, schema library, 3 healthcare schemas, unit tests

6. Phase 2: Agent Framework (Weeks 4-6)

Duration: 3 Weeks | Resources: 2 Engineers

Week 4: Orchestrator Agent

1. Setup LangGraph project structure
2. Define state machine (INIT → ANALYZE → EXTRACT → VALIDATE → COMPLETE)
3. Build transition logic with conditional branching
4. Implement checkpointing for recovery
5. Add error handling and retry policies
6. Create logging and tracing

Week 4-5: Analyzer Agent (1 VLM call/doc)

1. Create agent base class
2. Build classification prompt
3. Implement structure detection (tables, forms, sections)
4. Add page relationship logic
5. Build schema selector
6. Test with sample documents

Week 5: Extractor Agent (2 VLM calls/page)

1. Build extraction prompt with schema injection
2. Implement dual-pass extraction logic
3. Add field-by-field comparison
4. Create confidence scorer based on agreement
5. Add visual grounding
6. Handle table extraction

Week 6: Validator Agent (0-1 VLM call/doc)

1. Build schema validator with Pydantic
2. Add hallucination pattern detection
3. Implement cross-field consistency rules
4. Build cross-page merger
5. Add confidence calibration
6. Create output formatter (JSON, Markdown)

Deliverables: 4 working agents, LangGraph workflow, checkpointing, agent tests

7. Phase 3: Anti-Hallucination System (Weeks 7-8)

Duration: 2 Weeks | Resources: 2 Engineers

3-Layer Validation Framework

Layer	Technique	Owner	Implementation
1	Prompt Engineering	All Agents	Grounding rules, "do not guess", structured output
2	Dual-Pass Extraction	Extractor	Two passes, comparison, mismatch flagging
3	Pattern + Rule Validation	Validator	Repetition detection, cross-field, code checks

Confidence Score Actions

Score	Action
0.95+	Auto-Accept
0.85-0.94	Accept with note
0.70-0.84	Verify
0.50-0.69	Re-Extract
<0.50	Human Review

Target: Less than 10% of documents require human review

8. Phase 4: Integration & Testing (Weeks 9-10)

Duration: 2 Weeks | Resources: 2 Engineers

Week 9: REST API Development

- POST /api/v1/extract - Submit document
- GET /api/v1/tasks/{id} - Get status
- POST /api/v1/batch - Batch extraction
- GET /api/v1/health - Health check

Week 10: Testing Suite

- Unit tests (>80% coverage)
- Integration tests (end-to-end)
- Accuracy tests (golden dataset, 100+ docs)
- Adversarial tests (hallucination edge cases)
- CI pipeline setup

9. Phase 5: Deployment (Weeks 11-12)

Duration: 2 Weeks | Resources: 2 Engineers + DevOps

Week 11: HIPAA Compliance

- Verify 100% local processing (no external calls)
- Implement AES-256 encryption at rest
- Setup role-based access control
- Enable comprehensive audit logging
- Configure secure temporary file cleanup

Week 12: Production Launch

- Setup Prometheus/Grafana monitoring
- Configure alerting rules
- Write operations runbook
- Conduct team training
- Pilot testing and go-live

12. Success Metrics

Metric	Target
Field Extraction Accuracy	>95%
Hallucination Rate	<2%
Processing Speed	15-25 sec/page
VLM Calls per Page	3-4
System Uptime	>99.5%
Human Review Rate	<10%

11. Risk Management

Risk	Impact	Mitigation
Hallucinations	High	3-layer validation system
Poor doc quality	Medium	Enhancement pipeline
HIPAA violation	Critical	100% local processing
Model accuracy drift	Medium	Continuous monitoring
Hardware failure	Medium	Checkpointing, backups
Timeline overrun	Medium	2-week buffer