.NET Document Management System

Updated Project Plan and Implementation Timeline

Version 1.1 - February 19, 2025

Table of Contents

1. Project Overview

Project Objectives

- Develop an on-premises document management system
- Implement ML-based document classification
- Ensure secure document storage and retrieval
- Enable efficient search capabilities
- Provide document version control

Success Criteria

- System successfully deployed on internal infrastructure
- Document classification accuracy > 90%
- Search response time < 2 seconds
- System availability > 99.9%
- Full security compliance with internal standards

2. Project Timeline

Phase 1: Foundation and Infrastructure (Weeks 1-6)

1. Development Environment Setup (Week 1)
   - Install required software and tools
   - Configure development environments
   - Set up version control
   - Establish development standards
2. Infrastructure Setup (Weeks 1-3)
   - Deploy CI/CD infrastructure

   - Configure monitoring systems
   - Set up container registry
   - Establish backup systems

3. Database Foundation (Weeks 4-6)

   - Implement database schema
   - Create Entity Framework models
   - Set up maintenance procedures
   - Configure backup processes

Phase 2: Core Development (Weeks 7-14)

1. Core Application Architecture (Weeks 7-8)

   - Implement Clean Architecture layers
   - Set up identity solution
   - Configure authentication
   - Establish core services

2. Document Processing Services (Weeks 9-11)

   - Implement document handling
   - Set up storage systems
   - Create OCR services
   - Configure caching

3. ML.NET Integration and Model Development (Weeks 12-14)

   - Set up ML infrastructure
   - Develop classification models
   - Create training pipelines
   - Implement model management

Phase 3: API and Security (Weeks 15-18)

1. API Development (Weeks 15-16)

   - Create RESTful endpoints
   - Implement validation
   - Set up documentation
   - Configure rate limiting

2. Security Implementation (Weeks 17-18)

   - Configure authentication
   - Set up authorization
   - Implement audit logging
   - Establish security monitoring

Phase 4: Frontend and Testing (Weeks 19-24)

1. Frontend Development (Weeks 19-21)
    - Implement UI components
    - Create user interfaces
    - Set up real-time updates
    - Configure asset management
2. Testing Infrastructure (Weeks 22-24)
    - Implement testing frameworks
    - Create test suites
    - Perform load testing
    - Conduct security testing

Phase 5: Finalization (Weeks 25-28)

1. System Administration (Weeks 25-26)
    - Configure monitoring
    - Set up maintenance
    - Establish procedures
    - Implement health checks
2. Documentation (Weeks 27-28)
    - Create technical docs
    - Write user guides
    - Document procedures
    - Prepare training materials

3. Resource Requirements

Development Team

- 1 Solution Architect
- 2 Senior .NET Developers
- 1 ML/AI Developer
- 1 Frontend Developer
- 1 DevOps Engineer
- 1 QA Engineer

Infrastructure Requirements

- Development Environment
    - Developer workstations
    - Development server
    - Test environment
    - CI/CD server
- Production Environment

  - Application servers (4)
  - Database servers (2)
  - Storage servers (2)
  - Monitoring server

Software Requirements

- Development Tools
  - Visual Studio 2022
  - SQL Server 2022
  - ML.NET toolkit
  - Docker Desktop
  - Git
- Production Software
  - .NET 8.0 runtime
  - SQL Server 2022
  - MinIO
  - Jenkins/GitLab
  - Monitoring stack

4. Risk Management

Technical Risks

1. Performance
   - Document processing speed
   - ML model accuracy
   - System responsiveness
   - Storage capacity
2. Integration
   - Active Directory integration
   - Storage system compatibility
   - ML model deployment
   - Security implementation

Mitigation Strategies

1. Performance
   - Regular benchmarking
   - Performance testing
   - Capacity planning
   - Monitoring implementation
2. Integration

  - Early testing
  - Phased deployment
  - Fallback plans
  - Documentation

5. Quality Assurance

Testing Strategy

1. Continuous Testing

   - Unit testing
   - Integration testing
   - Performance testing
   - Security testing

2. Acceptance Criteria

   - Code coverage > 80%
   - API response time < 500ms
   - UI response time < 2s
   - Zero critical security issues

Monitoring

1. System Health

   - Server metrics
   - Application metrics
   - Database performance
   - Storage capacity

2. ML Performance

   - Model accuracy
   - Processing time
   - Resource usage
   - Error rates

6. Deliverables

Documentation

- System architecture documentation
- API documentation
- User guides
- Operations manual
- Security documentation

Software Components

- Backend API system
- Frontend application
- ML classification system
- Administration tools
- Monitoring dashboards

Support Materials

- Source code repository
- Test suites
- Deployment scripts
- Training materials
- Backup procedures

7. Success Metrics

Performance Metrics

- Document processing time < 5s
- Search response time < 2s
- Classification accuracy > 90%
- System uptime > 99.9%

Business Metrics

- Successful document processing
- User adoption rate
- System reliability
- Maintenance efficiency

8. Next Steps

1. Initiate development environment setup
2. Begin infrastructure deployment
3. Start database implementation
4. Commence core architecture development

Notes

- Regular progress reviews scheduled weekly
- Risk assessment updates bi-weekly
- Stakeholder updates monthly
- Team capacity reviews monthly