.NET Document Management System

Project Plan and Implementation Timeline

Version 1.0 - February 19, 2025

Table of Contents

1. Project Overview

Objectives

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

Success Criteria

- System deployed on internal infrastructure
- Document classification accuracy > 90%
- Search response time < 2 seconds
- High availability (99.9% uptime)
- Complete security compliance

2. Project Timeline

Phase 1: Foundation (Weeks 1-4)

1. Development Environment Setup (Week 1)
    - Install Visual Studio 2022
    - Configure SQL Server
    - Set up Git repository
    - Install ML.NET tools
    - Configure development machines
2. Database Implementation (Weeks 1-2)

- Port existing SQL schema
- Create Entity Framework models
- Implement migrations
- Set up stored procedures
- Configure indexing strategy

3. Core Architecture (Weeks 2-4)

- Create solution structure
- Implement Clean Architecture layers
- Set up dependency injection
- Configure logging infrastructure
- Implement authentication system

Phase 2: Core Development (Weeks 5-12)

1. Document Management (Weeks 5-7)

- Implement document storage
- Create version control system
- Develop metadata management
- Set up file system service
- Configure MinIO integration

2. ML.NET Implementation (Weeks 7-9)

- Set up ML.NET infrastructure
- Create classification models
- Implement training pipeline
- Develop evaluation system
- Create model management system

3. API Development (Weeks 9-12)

- Implement REST endpoints
- Create service layer
- Set up validation
- Implement error handling
- Create API documentation

Phase 3: Frontend and Integration (Weeks 13-16)

1. Frontend Development (Weeks 13-15)

- Set up Blazor WASM project
- Create component library
- Implement document viewer
- Develop search interface
- Create admin dashboard

2. Integration (Weeks 15-16)

    - Connect frontend and API
    - Implement real-time updates
    - Set up caching
    - Configure error handling
    - Implement monitoring

Phase 4: Testing and Deployment (Weeks 17-20)

1. Testing (Weeks 17-18)

    - Unit testing
    - Integration testing
    - Performance testing
    - Security testing
    - User acceptance testing

2. Deployment Preparation (Weeks 19-20)

    - Set up production environment
    - Configure monitoring
    - Implement backup systems
    - Create deployment scripts
    - Document procedures

3. Phase Details

Development Environment Setup

- Required Software Installation
    - Visual Studio 2022
    - SQL Server 2022
    - Git
    - ML.NET CLI
    - Docker Desktop
- Development Tools Configuration
    - Code analysis tools
    - Testing frameworks
    - CI/CD pipelines
    - Documentation generators

Database Implementation

- Schema Migration Tasks
    - Port existing tables
    - Create indexes
    - Set up constraints
    - Configure security
- Entity Framework Setup
    - Create domain models

    - Configure mappings
    - Set up migrations
    - Implement repositories

Core Architecture

- Solution Structure
    - Domain Layer
    - Application Layer
    - Infrastructure Layer
    - API Layer
    - Frontend Layer

Document Management

- Storage Implementation
    - File system structure
    - MinIO configuration
    - Backup system
    - Version control
- Metadata Management
    - Schema implementation
    - Search indexing
    - Classification storage
    - Audit logging

ML.NET Implementation

- Model Development
    - Feature engineering
    - Training pipeline
    - Model evaluation
    - Deployment system
- Integration
    - Document processing
    - Classification service
    - Performance monitoring
    - Model updates

4. Resource Requirements

Development Team

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

Infrastructure

- Development Servers
  - 2 Application Servers
  - 1 Database Server
  - 1 Storage Server
- Production Servers
  - 4 Application Servers
  - 2 Database Servers
  - 2 Storage Servers
- Development Tools
  - Visual Studio 2022 licenses
  - SQL Server 2022 licenses
  - Testing tools licenses

5. Dependencies

External Dependencies

- SQL Server 2022
- ML.NET Framework
- MinIO Object Storage
- Elasticsearch
- Redis Cache

Internal Dependencies

- Network Infrastructure
- Storage Systems
- Active Directory
- Backup Systems

6. Risk Management

Technical Risks

1. Performance
   - Document processing speed
   - Classification accuracy
   - Search response time
   - Storage capacity
2. Integration
   - System compatibility
   - Data migration
   - Security integration
   - Network configuration

Mitigation Strategies

1. Performance
    - Regular benchmarking
    - Optimization sprints
    - Capacity planning
    - Performance monitoring
2. Integration
    - Early testing
    - Phased deployment
    - Fallback plans
    - Documentation

7. Quality Assurance

Testing Strategy

1. Unit Testing
    - Domain logic
    - Services
    - Controllers
    - ML components
2. Integration Testing
    - API endpoints
    - Database operations
    - File operations
    - Classification system
3. Performance Testing
    - Load testing
    - Stress testing
    - Endurance testing
    - Scalability testing

Quality Metrics

- Code coverage > 80%
- Classification accuracy > 90%
- API response time < 500ms
- Search latency < 2s
- System uptime > 99.9%

8. Deliverables

Documentation

- Technical documentation
- API documentation
- User guides
- Deployment guides
- Training materials

Software Components

- Backend API
- Frontend application
- ML classification system
- Database schema
- Deployment scripts

Support Materials

- Source code
- Test suites
- Configuration files
- Training datasets
- Monitoring dashboards