Phase 6: Apex Programming

Overview

Phase 6 focuses on advanced Apex development to enhance system performance, implement complex business logic, and ensure robust error handling throughout the answer sheet evaluation system.

Key Components

6.1 Custom Apex Classes

Core Business Logic Implementation

- EvaluationManager: Main orchestration class for answer sheet processing
- AIEvaluationService: Handles AI integration with Google Gemini API
- DocumentProcessor: Manages document upload and processing
- OCRService: Integrates optical character recognition for scanned answer sheets
- AnalyticsHelper: Provides statistical analysis and reporting functions

6.2 Trigger Framework

Automated Record Processing

- AnswerSheetTrigger: Handles pre/post processing of answer sheet records
- MarkingSheetTrigger: Manages marking sheet creation and updates
- QuestionPaperTrigger: Validates question paper integrity
- BulkOperationTrigger: Processes bulk evaluation requests
- AuditTrailTrigger: Maintains comprehensive activity logging

6.3 SOQL/SOSL Optimization

Database Query Efficiency

- Selective Queries: Implemented with proper filtering and indexing
- Query Optimization: Reduced query rows and improved performance
- SOSL Integration: Full-text search for answer content analysis
- Aggregate Queries: Efficient data summarization for reporting
- Relationship Queries: Optimized parent-child record access

6.4 Asynchronous Processing

Scalable Background Operations

- Future Methods: Non-blocking API callouts to external services

- Queueable Jobs: Chained processing for complex evaluation workflows
- Batch Processing: Large-scale answer sheet evaluation handling
- Scheduled Jobs: Automated periodic tasks and maintenance
- Platform Events: Real-time event-driven architecture

6.5 Exception Handling

Robust Error Management

- Custom Exceptions: Domain-specific exception classes
- Try-Catch Blocks: Comprehensive error handling in all methods
- Rollback Mechanisms: Transaction integrity preservation
- Error Logging: Detailed logging for debugging and monitoring
- Graceful Degradation: System continues operating during partial failures

6.6 Testing Framework

Quality Assurance Implementation

- Unit Tests: 90%+ code coverage for all Apex classes
- Integration Tests: End-to-end workflow testing
- Mock Framework: Simulated external service responses
- Test Data Factory: Automated test data generation
- Performance Tests: Load testing for bulk operations

Advanced Features

6.7 Custom Metadata Types

Configuration Management

- Evaluation Settings: Configurable scoring parameters
- API Configuration: External service endpoint settings
- Business Rules: Dynamic rule engine configuration
- Feature Flags: Environment-specific feature toggles

6.8 Security Implementation

Enterprise Security Standards

- CRUD/FLS Enforcement: Proper field and object level security
- Sharing Rules: Record-level access control
- Governor Limits: Careful management of Salesforce limits
- Input Validation: Prevention of injection and malicious data

6.9 Performance Optimization

System Efficiency Enhancements

- Caching Strategy: Intelligent caching of frequently accessed data
- Lazy Loading: On-demand data retrieval
- Bulk Operations: Efficient processing of large datasets
- Memory Management: Optimal heap size utilization

Code Quality Standards

6.10 Development Best Practices

Professional Development Standards

- Naming Conventions: Consistent, descriptive naming
- Code Documentation: Comprehensive inline documentation
- Design Patterns: Implementation of proven patterns
- Code Reviews: Peer review processes
- Version Control: Proper branching and merging strategies

Implementation Status

- ✅ Custom Apex Classes: All core classes implemented and tested
- ✅ Trigger Framework: Comprehensive trigger system deployed
- ✅ SOQL/SOSL Optimization: Queries optimized for performance
- ✅ Asynchronous Processing: Future, Queueable, and Batch implemented
- ✅ Exception Handling: Robust error management throughout
- ✅ Testing Framework: 95%+ code coverage achieved
- ✅ Security Implementation: Enterprise security standards met
- ✅ Performance Optimization: System optimized for scale

Benefits

- Performance: Optimized for high-volume processing
- Reliability: Robust error handling and recovery
- Scalability: Handles large-scale evaluation operations
- Maintainability: Clean, well-documented, and tested code
- Security: Enterprise-grade security implementation
- Extensibility: Modular design for future enhancements