

WelVision Application

Complete UI & Backend Documentation

Generated on: June 10, 2025
Version: 1.0

Executive Summary

WelVision is a comprehensive industrial vision inspection application built with Python and Tkinter. The application provides advanced computer vision capabilities for quality control, user management, and real-time camera monitoring. It features a robust authentication system, role-based access control, and enterprise-grade security measures.

1. Application Architecture

1.1 Technology Stack

Component	Technology	Purpose
GUI Framework	Tkinter with ttk	Cross-platform desktop interface
Computer Vision	OpenCV + YOLO	Object detection and image processing
Database	MySQL	Data persistence and user management
Authentication	SHA-256 + Salt	Secure password hashing
Programming Language	Python 3.13	Core application development
Image Processing	PIL/Pillow	Image manipulation and display

2. Authentication System

2.1 Login Interface

- Clean, professional login window with company branding
- Email and password input fields with validation
- Secure password masking
- Account lockout protection after failed attempts
- Role-based redirection after successful authentication
- Real-time validation feedback

2.2 Security Features

- SHA-256 password hashing with unique salt per user
- Account lockout after 5 failed login attempts
- Session management and automatic logout
- Role-based access control (User, Admin, Super Admin)
- Audit logging for all authentication events
- Input sanitization to prevent SQL injection

2.3 User Roles

Role	Permissions	Access Level
User	View cameras, basic operations	Limited
Admin	User management, advanced features	Extended
Super Admin	Full system access, user creation	Complete

3. Main Application Interface

3.1 Application Layout

The main application uses a tabbed interface design with a professional dark theme. The interface is optimized for industrial environments with high contrast and clear visibility. Navigation is intuitive with role-based tab visibility.

3.2 Application Tabs

■ Camera Tab

Frontend Features: • Dual camera feed display (Object Detection and Barcode) • Real-time YOLO object detection overlay • Live camera status indicators • Image capture and save functionality • Detection results display panel **Backend Integration:** • OpenCV camera interface with error handling • YOLO model loading and inference • Real-time frame processing in separate threads • Automatic camera reconnection on failure • Image preprocessing and enhancement

■ Inspection Tab

Frontend Features: • Product selection via dropdown menu • Manual image upload and processing • Results visualization with annotations • Quality metrics display • Pass/Fail determination interface **Backend Integration:** • Image analysis algorithms • Product specification database lookup • Quality threshold comparison • Automated report generation • Historical data logging

■ Reports Tab

Frontend Features: • Date range selection for report generation • Multiple report format options • Data visualization charts and graphs • Export functionality (PDF, Excel) • Real-time statistics dashboard **Backend Integration:** • Database query optimization • Statistical analysis algorithms • Report template engine • Data aggregation and filtering • Automated scheduling system

■■ Settings Tab

Frontend Features: • System configuration panels • Camera calibration interface • Threshold adjustment controls • User preference settings • System diagnostics display **Backend Integration:** • Configuration file management • System parameter validation • Hardware communication protocols • Backup and restore functionality • Performance monitoring

4. User Management System

4.1 User Management Tab

Frontend Features: • Two-panel layout with user list and details form • Real-time search and filtering capabilities • Role-based access control interface • Password change dialogs with validation • Bulk user operations support **Backend Integration:** • MySQL database with optimized queries • Secure password hashing and salt generation • User session management • Audit trail logging • Input validation and sanitization

4.2 CRUD Operations

Operation	Frontend Interface	Backend Process
Create	New user form with validation	Password hashing, database insertion
Read	User list with search/filter	Optimized database queries
Update	Editable user details form	Validation, secure updates
Delete	Confirmation dialog	Cascading deletes, audit logging

5. Database Schema

5.1 Users Table

Table: users • id (INT, PRIMARY KEY, AUTO_INCREMENT) • employee_id (VARCHAR, UNIQUE, NOT NULL) • email (VARCHAR, UNIQUE, NOT NULL) • password_hash (VARCHAR, NOT NULL) • salt (VARCHAR, NOT NULL) • role (ENUM: 'User', 'Admin', 'Super Admin') • first_name (VARCHAR) • last_name (VARCHAR) • is_active (BOOLEAN, DEFAULT TRUE) • failed_login_attempts (INT, DEFAULT 0) • last_login (TIMESTAMP) • created_at (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP) • updated_at (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP ON UPDATE)

6. Advanced System Features

6.1 Error Handling

• Comprehensive exception handling throughout the application • User-friendly error messages and validation feedback • Automatic error logging and debugging information • Graceful degradation for hardware failures • Recovery procedures for database connection issues

6.2 Performance Optimization

• Multi-threaded camera processing for smooth real-time display • Optimized database queries with proper indexing • Efficient memory management for image processing • Lazy loading for large datasets • Caching mechanisms for frequently accessed data

6.3 Security Measures

• SQL injection prevention through parameterized queries • Input validation and sanitization • Secure password storage with SHA-256 and salt • Session timeout and automatic logout • Role-based access control enforcement • Audit logging for all critical operations

7. Deployment and Maintenance

7.1 System Requirements

Software Requirements: • Python 3.13 or higher • MySQL 8.0 or higher • OpenCV 4.x • Tkinter (included with Python) • Required Python packages (see requirements.txt) **Hardware Requirements:** • Minimum 8GB RAM (16GB recommended) • Multi-core processor (Intel i5 or equivalent) • USB 3.0 ports for camera connections • Minimum 1920x1080 display resolution

7.2 Installation Guide

1. Install Python 3.13 and required dependencies 2. Set up MySQL database and create welvision_db 3. Run database initialization scripts 4. Configure camera connections and settings 5. Execute main.py to launch the application 6. Complete initial user setup and configuration

8. Future Enhancement Opportunities

• Web-based interface for remote access • Advanced AI models for defect detection • Integration with enterprise ERP systems • Mobile application for notifications • Advanced analytics and machine learning • Cloud deployment and scalability options • API development for third-party integrations

9. Conclusion

The WelVision application represents a comprehensive solution for industrial vision inspection with robust user management capabilities. The system demonstrates enterprise-grade security, scalable architecture, and user-friendly interface design. The modular structure allows for easy maintenance and future enhancements while maintaining high performance and reliability.