CleanShare

Quality Assurance Test Plan

1. Overview

This Quality Assurance test plan shows how testing will be done for our application **CleanShare**, an offline desktop application that detects alcoholic beverages in images and automatically blurs them.

Main Workflow

1. Load a JPEG/PNG image.
2. Run the Detection Engine (ML model) to find alcoholic beverages.
3. Use the Redaction Pipeline to create masks and apply blur.
4. Allow the user to preview and export the redacted image.

Core Components Under Test

- **IOService** – loads/saves images and checks paths and formats.
- **Detection Engine** – preprocesses images, runs the ML model, outputs detections.
- **Redaction Pipeline** – builds masks and applies blur to detected areas.
- **Integration Testing** – the full workflow of the application.

2. Test Objectives

- Verify loading and saving of valid JPEG/PNG images.
- Ensure invalid paths, formats, or corrupted files are handled safely.
- Confirm Detection Engine outputs are structured well.
- Confirm Redaction Pipeline only alters or blurs pixels inside the detected region.
- Make sure the workflow operates correctly without crashes.

3. Test Approach

Unit Testing

IOService

- Valid PNG/JPEG loads successfully.
- Non-existent path triggers an error and is handled correctly.

Detection Engine

- Initializes successfully.
- Returns good detections (bounding boxes around alcohol and a score).
- Image quality is not greatly altered.

Redaction Pipeline

- Applies blur only inside detected areas.
- Output image dimensions match input.
- If there is no detection, then no blur is applied.

Integration Testing

Image with alcohol:
Load image → detect → blur → export; blurred regions appear in correct locations.

Image without alcohol:
Pipeline runs, output is visually identical to input.

Invalid input:
Missing file or bad extension produces an error with no crash.

4. Testing Requirements

Before testing begins:

- Core modules compile.
- Basic test images are available and work.
- ML model is working along with other external resources.

For testing to be considered complete:

- All planned unit and integration tests work.

- High severity errors are fixed and / or documented.
- Main workflow passes on test images.
- No crashes or data corruption occur on valid inputs.