CISC 320 Requirements Analysis Document
CleanShare

Section 3 - Group 6
David Balan
Anthony Grecu
Charlie Guarasci
Liam Harper-Mccabe
Kieron Luke
Mark Nistor
Kayetan Protas

November 13, 2025

Contents

1 Executive Summary

In today's digital age, maintaining a professional and polished social media presence is essential for our personal image, career opportunities, and overall reputation—especially with the rise of cancel culture and consequences for poor choices made online. So, meet CleanShare: an innovative web application designed to help users share photos that include certain adult beverages they may not want online.

CleanShare allows users to simply drag and drop their .jpg or .png images into CleanShare's UI, where, with a click of a button, any alcoholic beverages will be automatically blurred. Whether you're a Queen's student worried about maintaining a polished image for prospective employers or a professor wanting to keep their social life private, CleanShare allows you to share photos online without the worry of blemishing your image.

2 Background and History

Universities maintain strict guidelines around public representation, especially concerning alcohol in the media. At Queen's University, many student clubs and organizations promote events through social media. However, the presence of visible alcoholic beverages in photos can violate student conduct policies and damage reputations. Currently, such images must be manually edited—a process that is slow, inconsistent, and dependent on human judgment.

CleanShare addresses this problem by providing a locally executed desktop application that detects and blurs alcoholic beverages in photos automatically. The tool combines traditional computer vision with a compact convolutional neural network (CNN) optimized for on-device inference. Detection is based on the Liquor Data dataset from Lamar University (Roboflow: `https://universe.roboflow.com/lamar-university-venef/liquor-data`), which contains labeled images of alcoholic containers (beer, wine, and spirits) suitable for object detection model training. By running fully offline, CleanShare ensures privacy and reliability while maintaining compatibility with university IT policies that restrict cloud-based content processing.

The project prioritizes transparent, explainable functionality over black-box automation. Detection performance, model reliability, and processing latency are treated as measurable system attributes rather than secondary effects of implementation. The design goal is not aesthetic enhancement but compliance automation—reducing manual editing without compromising privacy or accuracy.

3 Purpose and Scope

The purpose of CleanShare is to develop an efficient, privacy-respecting desktop tool that automatically detects and blurs alcoholic beverages in digital photographs before sharing. It enables students, clubs, and campus organizations to prepare "safe-to-share" media without manual redaction or third-party upload, aligning with institutional image and conduct policies.

3.1 System Scope

Automatic detection and blurring: Alcoholic containers are identified using a hybrid detection pipeline combining OpenCV contour and color feature extraction with CNN-based classification via ONNX Runtime.

Local inference: All computation runs on the user's machine using pre-trained ONNX weights exported from a YOLOv8-based model fine-tuned on the Liquor Data dataset.

Platform target: Windows 10+ desktop systems with C++17 or later, Qt 6.x, and OpenCV 4.x.

Hardware constraints: Minimum 8 GB RAM, CPU-only inference under 10 s per 1080p image. GPU acceleration (CUDA) optional.

3.2 Assumptions and Dependencies

Functional Qt and OpenCV libraries installed in the runtime environment. Users provide static image files; video or batch modes are out of scope for the current release.

3.3 Validation and Evaluation

System evaluation uses a held-out subset of Liquor Data annotated images. Performance metrics include: Precision/Recall and F1 score to measure detection correctness. Mean Average Precision (mAP@0.5) as the principal accuracy indicator. False positive rate against non-alcoholic container images (e.g., soda cans). A successful implementation must achieve $\geq$ 80% recall, $\leq$ 15% false positives, and average processing time $\leq$ 10 s for 1080p input.

3.4 Stakeholders

Primary: University students, club social media coordinators, campus event organizers. Secondary: Faculty communications staff, student services offices, and event photographers.

4 Objectives and Success Criteria

The objective of CleanShare is to provide a privacy-preserving desktop application that automatically detects and blurs alcoholic beverages in images before sharing. The system will allow users, particularly students and campus organizations, to safely post photos online without revealing intoxicating beverages that may conflict with institutional standards or professional expectations. The software must run entirely on the user's computer, ensuring no compromising images or personal data leave the device.

The minimum viable product will enable users to load an image, automatically detect and blur alcoholic beverages using a pre-trained model, view a preview of the processed image, make optional manual adjustments to blur regions, and export the final blurred image at its original resolution. The interface should support basic user workflow from import to export with no network connection required. The application will use Qt for the interface, OpenCV for preprocessing and redaction, and ONNX Runtime for model inference.

Success will be measured by detection accuracy and system performance. The detection model must achieve at least 80 percent recall when evaluated on a held-out subset of labeled test images, while also maintaining a low false positive rate. The application must process a 1080p image in 10 seconds or less on typical laptop hardware without GPU acceleration. All processing will occur locally, and no user images will be transmitted or stored beyond temporary in-memory data. Manual editing tools must allow users to correct missed detections before exporting. The exported images must match the original resolution and contain all selected blur regions. If no alcohol is detected, the system will notify the user and still allow manual marking and export. Once these core requirements are achieved, additional improvements such as pixelation redaction options, performance optimizations, or batch processing may be considered for future releases.

5 Definitions and Acronyms

CleanShare is a Windows desktop application that automatically detects and blurs alcoholic beverages in photos so they're safe to share. It processes images locally (no cloud), using a C++/Qt GUI, OpenCV for image handling, and a bundled CNN served via ONNX Runtime for robust detection. Core user features include image upload, automatic detection+blur, side-by-side preview, manual correction tools, and export at the original resolution. Target performance is $\leq$ 10 s per 1080p photo with $\geq$ 80% recall and low false positives.

5.1 Primary modules (logical view)

- Presentation (Qt GUI): File open/save, side-by-side preview, blur-strength controls, and manual tools (brush/rectangle) to add/remove regions.
- Application Core: Orchestrates the workflow (import $\rightarrow$ detect $\rightarrow$ blur $\rightarrow$ preview $\rightarrow$ export), holds session state and configuration (e.g., detection threshold, blur type).

- Detection Engine: Image pre-processing → CNN inference (ONNX Runtime) → post-processing (NMS, thresholding). Packaged, pre-trained CNN weights enable local inference. A simple OpenCV heuristic may exist as a baseline/fallback.
- Redaction Pipeline: Builds ROI masks from detections and applies blurring while preserving original dimensions on export; future variants (pixelation/mosaic) are optional extensions.
- I/O & Privacy: Supports JPEG/PNG input and exports blurred images; only ephemeral metadata (boxes/masks) is held in memory; no images or personal data leave the device.
- Evaluation (dev utility): Dataset-based checks for recall/false positives and timing against the success criteria.

6 General Description of the System

Primary use case: "Blur a photo and export"

1. Launch & Idle: App starts → main window (no network required). Platform target: Windows.
2. Import: User selects an image (JPEG/PNG). App validates format, loads pixels, records original dimensions.
3. Pre-process for Detection: Convert to detector input (resize/normalize); keep a mapping to original coordinates.
4. Detect Alcohol Containers: Run CNN (ONNX Runtime). Optionally fuse heuristic detection; perform NMS and confidence thresholding.
5. Post-process Detections: Map boxes/masks back; build ROI mask.
6. Apply Redaction: Apply blur (default Gaussian) to ROI mask.
7. Preview & Manual Adjustments: Show side-by-side; user may edit blur regions.
8. Export: Write blurred image; clear metadata.

6.1 Alternate flows & error handling

- No detections: Notify "No alcohol detected." Allow manual marking.
- Invalid file: Show an error and return to Import.
- Slow inference: Show progress and allow cancel; fallback heuristic.
- Privacy: No images or user data are sent to any server.

6.2 (Optional) Batch flow (future extension)

User selects a folder → app iterates the pipeline per image, applies consistent settings, and writes outputs to an export folder.

7 Program Flow

7.1 Program Flow Diagram

The program flow diagram illustrates the overall execution sequence of CleanShare — from user input to image export. It shows how user actions (such as importing an image) interact with system processes like preprocessing, detection, blurring, and export. This helps visualize the step-by-step control flow and major function calls within the application.

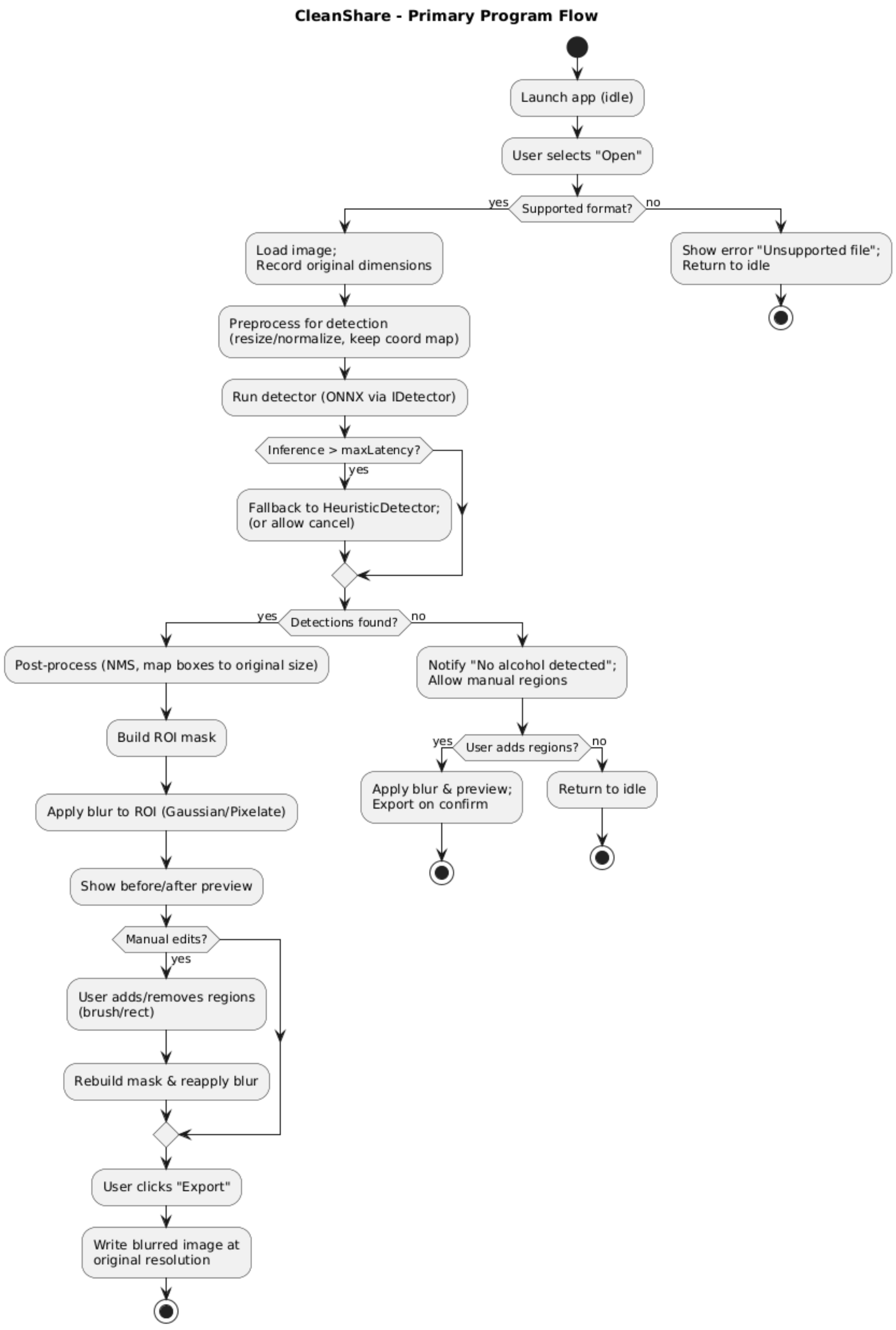

CleanShare - Primary Program Flow
Launch app (idle)
User selects "Open"
Supported format?
yes
no
Load image;
Record original dimensions
Show error "Unsupported file";
Return to idle
Preprocess for detection
(resize/normalize, keep coord map)
Run detector (ONNX via IDetector)
Inference > maxLatency?
yes
Fallback to HeuristicDetector;
(or allow cancel)
Detections found?
yes
no
Post-process (NMS, map boxes to original size)
Notify "No alcohol detected";
Allow manual regions
Build ROI mask
User adds regions?
yes
no
Apply blur & preview;
Export on confirm
Return to idle
Apply blur to ROI (Gaussian/Pixelate)
Show before/after preview
Manual edits?
yes
User adds/removes regions
(brush/rect)
Rebuild mask & reapply blur
User clicks "Export"
Write blurred image at
original resolution

Figure 1: Program Flow Diagram for CleanShare

8 Functional Requirements

- Allows users to upload images in JPEG, PNG.
- Verifies the file type and size before processing.
- Detect and blur alcoholic beverages automatically using CNN (ONNX Runtime).
- Apply non-maximum suppression (NMS) and confidence thresholding.
- Identifies bounding boxes or regions and applies Gaussian blur.
- Offer preview mode with side-by-side comparison.
- Allow users to manually correct using brush/rectangle tools.
- Allow exporting at original resolution.
- Handle failed uploads and model inference errors gracefully.
- Ensure all processing remains local with no data leaving the device.

9 Non-Functional Requirements

9.1 Performance requirements

- Process a 1080p image in ≤ 10 s.
- CNN detection recall $\geq 80\%$, low false positives.
- GUI response time $\leq$300 ms.

9.2 User experience

- Workflow intuitive for first-time users.
- Consistent layout, labeled controls.
- All major operations in ≤ 3 user actions.
- Visual feedback for operations.
- Undo/redo supported without corruption.
- High-contrast actionable buttons.

9.3 Ethical

- Evaluate model for bias.
- Notify users about false positives.

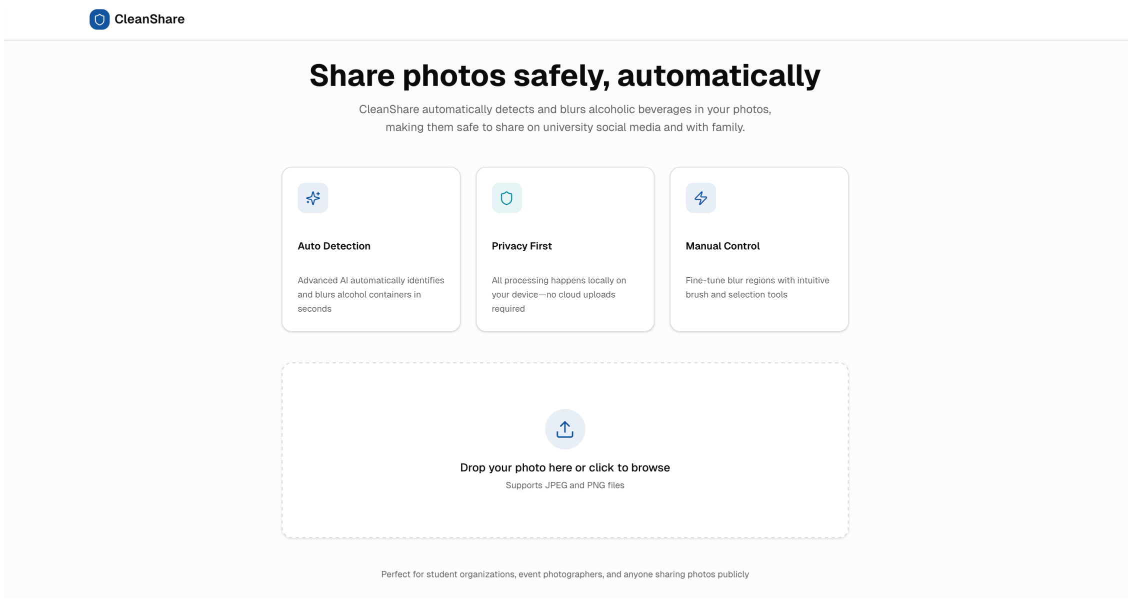
9.4 Documentation

- User manual with installation, usage, troubleshooting.
- Developer guide with architecture and build process.
- CNN model and preprocessing pipeline versioned for reproducibility.

10 Layout of GUI

10.1 Landing Page

Serves as the user's entry point to the application. Features a central area for uploading images via drag-and-drop or browse. Displays core features: Auto Detection, Privacy First, and Manual Control.



CleanShare

Share photos safely, automatically

CleanShare automatically detects and blurs alcoholic beverages in your photos, making them safe to share on university social media and with family.

Auto Detection

Advanced AI automatically identifies and blurs alcohol containers in seconds

Privacy First

All processing happens locally on your device—no cloud uploads required

Manual Control

Fine-tune blur regions with intuitive brush and selection tools

Drop your photo here or click to browse

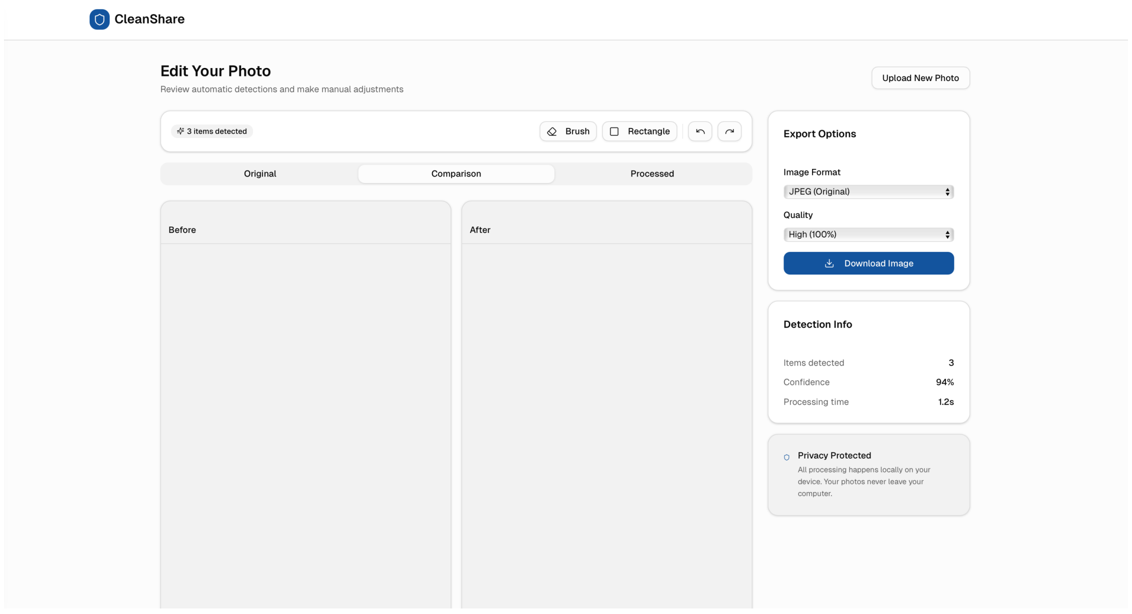
Supports JPEG and PNG files

Perfect for student organizations, event photographers, and anyone sharing photos publicly

Figure 2: CleanShare Home Page Interface

10.2 Edit/View Page

Loads after processing. Side-by-side comparison of before/after. Right sidebar provides export options, file type, insights, and privacy policy. Users can download or upload a new image directly.



CleanShare

Edit Your Photo

Review automatic detections and make manual adjustments

Upload New Photo

3 items detected

Brush | Rectangle

Original | Comparison | Processed

Before

After

Export Options

Image Format: JPEG (Original)

Quality: High (100%)

Download Image

Detection Info

Items detected	3
Confidence	94%
Processing time	1.2s

Privacy Protected

All processing happens locally on your device. Your photos never leave your computer.

Figure 3: CleanShare Edit/View Interface

11 Assumptions and Constraints

We assume all members attend meetings, communicate changes, and complete tasks on time. Users will use JPEG/PNG files as input.

Constraints include limited timeframe restricting testing and development, learning new technologies like OpenCV, and the need to run offline without cloud APIs.

12 Roles of the Team

Team Member(s)	**Role**	**Responsibilities**
Anthony Grecu & David Balan	Backend Developers	Handle image processing and develop machine learning model.
Mark Nistor & Liam Harper-Mccabe	System Engineers	Integrate the back end and front end systems.
Kieron Luke & Charlie Guarasc	Front End Developers	Design and create the GUI.
Kayetan Protas	Documentation and Quality Assurance Engineer	Document the project and create test scripts.
Every team member assumes a coding role.		

Table 1: Roles and Responsibilities of the CleanShare Team

13 System Diagrams

13.1 Logical View Diagram

The logical view diagram presents the main architectural components of CleanShare and their relationships. It shows how modules such as the GUI, Application Core, Detection Engine, and Redaction Pipeline interact logically. This diagram provides a high-level understanding of the system's structure and separation of responsibilities.

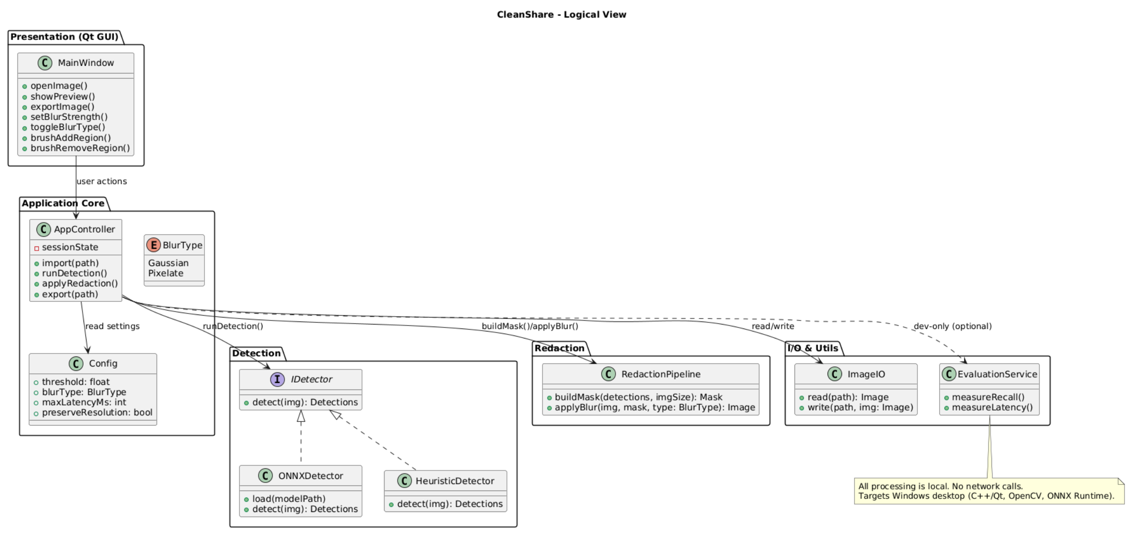


Figure 4: Logical View Diagram of CleanShare Architecture