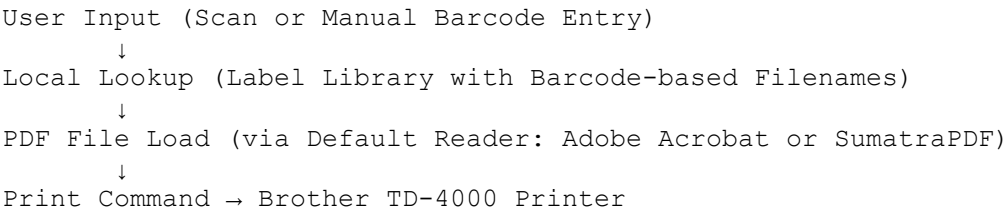


Technical Specifications Document Barcode Label Printing Program (v0.1.0)

1. Overview

The Barcode Label Printing Program is a lightweight, Python-based desktop application developed to streamline the printing of labels for house-packaged items. Label files are exported from the internal Menu Builder – Print Shop platform and stored locally for direct access by the application. Users can scan or enter a barcode, retrieve the corresponding label, and send it to a thermal printer with minimal steps. Version 0.1.0 focuses on local execution, reducing workflow delays and building a scalable foundation for future enhancements including server-based access and kiosk deployment.

2. System Architecture



All operations are local in v0.1.0. Future versions will support remote database querying and web-kiosk integration.

3. Dependencies

Type	Package / Component	Notes
Language	Python 3.11+	Uses tkinter, subprocess, os, etc.
GUI	tkinter	Standard library; no extra install needed
PDF Tool	Adobe Acrobat Reader OR SumatraPDF	Customizable via config
Print	Windows default printer interface	Preconfigured to Brother TD-4000
Other	PyPDF2	Planned for future parsing and control

4. Operating System Compatibility

OS	Supported	Notes
Windows	v0.1.0	Fully supported; tested on Windows 10+
Linux	Planned	CLI/daemon version targeted in v1.0.0
macOS	Not supported	No support planned at this stage

5. Label Library System

- Purpose: Pre-exported labels for house-packaged food
 - Source: All label files are generated through the Menu Builder – Print Shop and manually saved using the defined barcode-based naming convention, then placed into the local label directory
 - Naming Convention: BARCODE Name of Item.pdf
Example: SAGEMB000000204 Chicken Salad Bento Box.pdf
 - Storage Location: User-defined local folder (configurable via settings)
 - File Format: .pdf only (1 label per page, vector-safe)
 - Lookup Logic: Barcode input triggers direct filename match and opens the corresponding label for print
-

6. Configuration & Customization

- Config File: .ini file planned for v0.1.1+ release
 - Customize paths to PDF reader, label directory, and printer
 - Default Reader: Supports Adobe Acrobat Reader or SumatraPDF
 - Fallback Handling: Planned improvements include dialogs for missing files and error logging
-

7. Printer & Format Requirements

Device	Specification
Model	Brother TD-4000
Connection	USB (Windows default printing)
Supported Sizes	4" x 2" / 4" x 6" (configurable)
PDF Requirements	Vector-safe PDF, 1 label per page

8. Security & Access Control

- No network operations or external file transfers in v0.1.0
 - All execution is local and sandboxed to the host PC
-

9. Known Limitations

- Label files must be manually named and placed in the folder
 - No error dialog for missing or mismatched barcodes in v0.1.0
-

10. Development Roadmap

v0.1.0 – Prototype Release

- Local-only label lookup and printing
 - GUI with barcode entry, quantity selection, and print trigger
 - Uses Adobe Acrobat or SumatraPDF for PDF rendering
 - Hardcoded paths (manual edits only)
 - Windows-only compatibility
-

v0.1.1 – Configuration Refactor

- Implement .ini config file for:
 - Reader path
 - Label folder
 - Default printer
 - Graceful error messages for missing files
 - File existence validation before print
 - Code cleanup and modularization
-

v0.1.5 – Installer & Label Directory Selection

- Windows installer packaging using PyInstaller + Inno Setup
 - Prompts user to select label directory or creates a default on first run
 - Installer writes selection to the .ini configuration file introduced in v0.1.1
 - Program always reads paths from the config file at runtime
 - Improved accessibility for non-technical users
-

v0.2.0 – Linux CLI Compatibility & Kiosk Integration

- Develop CLI version of the program with barcode and quantity as command-line arguments
 - Enable silent print execution without GUI dependencies
 - Compatible with Fedora-based kiosks running a company-specific web interface
 - Supports shell calls or backend API triggers from the web front-end
 - Basic stdout logging for print success/failure
 - Foundation for future Linux print daemon (planned for v1.0.0)
-

v0.2.5 – Remote Label Sync via Menu Builder Integration

- Integrate with company Menu Builder – Print Shop system
- Allow venue managers to flag recipes in their quarterly menu as “label-ready”
- Backend generates a venue-specific label library using standardized filenames
- Labels hosted on the company server with per-venue access
- Program syncs to the local label folder via API (manual or scheduled pull)
- Implements smart fallback logic as follows:
 - Primary: Use local label file if available and valid
 - Secondary: If file missing or corrupted, fetch from the server-hosted label library
 - Tertiary: If file is not found server-side, trigger generation of a fresh label from the Menu Builder – Print Shop system

Note: This milestone is coordinated with a side-branch development effort on the Menu Builder platform to add labeling flags, automated label generation, and a secure venue-level API.

v1.0.0 – Full Operational Release

- Finalize and stabilize all features introduced in prior milestones
- Refactor GUI (where applicable) to include optional embedded label preview pane
- Expand local folder structure handling to support:
 - Archived vs active label sets
 - Optional subfolders by item category or service type
- Add optional sync scheduling (e.g., at startup or set interval)
- Improve update/deployment mechanisms (silent updates or installer refresh)
- Consolidate error handling, logging, and fallback modes into robust service logic (*Fallback logic is now defined in v.0.2.5; v1.0.0 handles refinement and resilience framework*)
- Tag and version final release as kiosk- and production-ready

11. Feature Plan

Core Functional Features

- Barcode-based label lookup via scan or manual input
 - Print quantity control from simple GUI or CLI
 - Windows GUI and Linux CLI compatibility
 - Local label file handling with standardized filename structure
 - Smart fallback logic:
 - Use local label if valid
 - Pull from server if missing or corrupted
 - Generate from Menu Builder if unavailable
 - Configurable paths (PDF reader, label folder, printer) via .ini
-

Automation & Integration

- Venue-specific label selection from Menu Builder (label-ready flag)
 - API integration for label sync by venue
 - Initial folder creation and config generation via installer
 - CLI support for headless kiosk execution
 - Server-side label repository maintained per venue
-

Operational Enhancements (Planned for v1.0.0+)

- Embedded label preview pane in desktop GUI
 - Support for organized subfolders (e.g., by category or item type)
 - Scheduled sync option (on startup or timed interval)
 - Error logs and print activity logs
 - Optional silent updates or reinstallation framework
-

Deferred or Optional Features (Post-1.0 Consideration)

- Batch print queue (e.g., print multiple barcodes in sequence)
- Label version tracking (e.g., date-stamped or versioned files)
- Manual lookup by item name through browsing organized subfolders (for use cases without barcode access)