

---

# **Software Requirements Specification**

**for**

# **MerchTools**

**Version 1.0**

**Prepared by Charles Clark**

**2 November 2025**

## Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>Revision History .....</b>	<b>iii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose .....	1
1.2 Document Conventions .....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Project Scope .....	2
1.5 References .....	2
<b>2. Overall Description .....</b>	<b>2</b>
2.1 Product Perspective .....	2
2.2 Product Features .....	3
2.3 User Classes and Characteristics .....	3
2.4 Operating Environment .....	4
2.5 Design and Implementation Constraints.....	4
2.6 User Documentation .....	4
2.7 Assumptions and Dependencies .....	4
<b>3. System Features .....</b>	<b>5</b>
3.1 Scan or Search Item.....	5
3.1.1 Description and Priority .....	5
3.1.2 Stimulus/Response Sequences .....	5
3.1.3 Functional Requirements.....	5
3.2 Record Counts, Notes, and Photos .....	6
3.2.1 Description and Priority .....	6
3.2.2 Stimulus/Response Sequences .....	6
3.2.3 Functional Requirements.....	6
3.3 Generate PDF Report.....	7
3.3.1 Description and Priority .....	7
3.3.2 Stimulus/Response Sequences .....	7
3.3.3 Functional Requirements.....	7
3.4 Email/Share Report .....	8
3.4.1 Description and Priority .....	8
3.4.2 Stimulus/Response Sequences .....	8
3.4.3 Functional Requirements.....	8
3.5 History and Export .....	8
3.5.1 Description and Priority .....	8
3.5.2 Stimulus/Response Sequences .....	9
3.5.3 Functional Requirements.....	9
3.6 Add SKU .....	9
3.6.1 Description and Priority .....	9
3.6.2 Stimulus/Response Sequences .....	9
3.6.3 Functional Requirements.....	10
<b>4. External Interface Requirements .....</b>	<b>10</b>
4.1 User Interfaces.....	10
4.2 Hardware Interfaces.....	10
4.3 Software Interfaces.....	10
4.4 Communications Interfaces .....	11
<b>5. Other Nonfunctional Requirements.....</b>	<b>11</b>
5.1 Performance Requirements.....	11
5.2 Safety Requirements.....	11
5.3 Security Requirements.....	11
5.4 Software Quality Attributes.....	12
<b>6. Other Requirements .....</b>	<b>12</b>

<b>Appendix A: Glossary.....</b>	<b>14</b>
<b>Appendix B: Analysis Models .....</b>	<b>14</b>
<b>Appendix C: Issues List.....</b>	<b>14</b>

## Revision History

Name	Date	Reason For Changes	Version
Charles Clark	11/2/2025	First complete SRS for submission	1.0
Charles Clark	11/8/2025	Added feature <a href="#">3.6</a> Add SKU	1.1

# 1. Introduction

## 1.1 Purpose

This software requirements specification (SRS) document defines MerchTools v0.1.0, a mobile application designed to aid with shelf audits and inventory management in the retail space. The features of MerchTools will enable field associates to identify products (e.g., SKUs)—through scanning or searching—record counts and notes, attach photos, and generate a timestamped PDF report for share/email. This SRS covers the mobile client only (Android). It does not specify any server or cloud components; future versions may add those in separate documents.

## 1.2 Document Conventions

- **Semantics:**
  - **Shall** = mandatory
  - **Should** = recommended
  - **May** = optional
- **Requirement IDs:**
  - **Functional** = FR-x.y
  - **Nonfunctional** = NFR-x.y
- **Priorities** are not inherited; each requirement states its own priority:
  - **P1** (must for v0.1.0)
  - **P2** (nice to have for v0.1.x)
  - **P3** (future)
- **Acceptance criteria** accompany key features to aid verification.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for all readers interested in MerchTools; however, the primary intended audience includes instructors and reviewers, developers/testers, and stakeholders (e.g., store/Pepsi managers, associates, and vendor reps). The following text outlines reading suggestions for each of the intended audience:

- **Instructors/Reviewers:** Sections 1-2 for scope and context; Sections 3 & 5 for core behavior and quality requirements.
- **Developers/Testers:** Sections 1-5 for scope, context, system features, and requirements; Appendices B (models) & C (open issues) as added.

- **Stakeholders:** Sections 2.1-2.4 (relevant overview) and Section 3 summaries.

Refer to **Appendix A** for key term definitions.

## 1.4 Project Scope

This SRS specifies the initial release (v0.1.0) of MerchTools as a standalone, offline-first Android application. The purpose of this product is to reduce out-of-stock (OOS) impact—such as lowering consumer satisfaction and lost sales—and time spent documenting shelf conditions and correcting inventory counts compared to the current practice. MerchTools unify scanning/searching, note taking, photo capture, and PDF reporting in a single, simple workflow. Business goals: faster escalation and correction of inventory/shelf issues, clear accountability, and portable reports that optimize audit processes.

## 1.5 References

- Clark, Charles. (2025). *MerchTools – Project Proposal* (Oct 24, 2025).
- Google Developers. “Barcode Scanning | ML Kit.” *Google Developers*, [developers.google.com/ml-kit/vision/barcode-scanning](https://developers.google.com/ml-kit/vision/barcode-scanning).
- Anthony. “Integrating Google ML Kit for Barcode Scanning in Jetpack Compose Android Apps.” *Medium*, ProAndroidDev, 11 Jan. 2025, [proandroiddev.com/integrating-google-ml-kit-for-barcode-scanning-in-jetpack-compose-android-apps-5deda28377c9](https://proandroiddev.com/integrating-google-ml-kit-for-barcode-scanning-in-jetpack-compose-android-apps-5deda28377c9). Accessed 3 Nov. 2025.
- “Material Design 3 in Compose | Jetpack Compose.” *Android Developers*, [developer.android.com/develop/ui/compose/designsystems/material3](https://developer.android.com/develop/ui/compose/designsystems/material3).
- “Develop for Android - Core Areas.” *Android Developers*, 2024, [developer.android.com/develop#core-areas](https://developer.android.com/develop#core-areas).
- Internal observation notes & short interviews with store and PepsiCo personnel (to be summarized in Appendix B).

## 2. Overall Description

### 2.1 Product Perspective

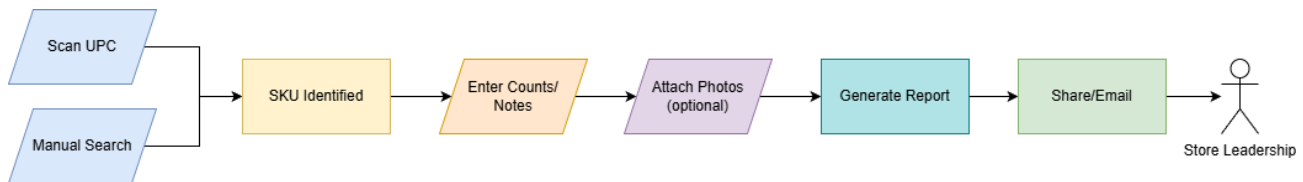
MerchTools is a new, self-contained product. It does not replace existing store systems; instead, it complements them by bridging the gap between internal and external inventory auditing and control through producing standardized PDF and optional CSV outputs that any recipient can read. External touchpoints are limited to the device camera (for scanning/photos) and the Android Sharesheet/email apps for sending reports.

**High-level components (conceptual):**

- **Scanner & Lookup** (camera barcode/UPC scan or manual search)
- **Add SKU** (add SKU to catalog)
- **Audit Capture** (counts, notes, photos)
- **Report Builder** (PDF generation and optional CSV)
- **Share/History** (email/share and past audits)

**2.2 Product Features**

- Scan a UPC/barcode or search a product to identify the correct SKU.
- Enter counts and short notes with validation guardrails.
- Attach photos as evidence (optional).
- Generate a clear, timestamped PDF with a cover summary and item details.
- Email/share the report; see past reports and export CSV.

**2.3 User Classes and Characteristics**

- **Pepsi Field Associate (e.g., merchandiser, sales rep, etc.) (primary):** Frequent use—multiple uses per week to daily; needs speed and efficiency; minimal typing; varying technical expertise (baseline technical comfort is established through the company mobile application used for daily work).
- **Store / Department Leadership (recipient):** Consumes PDF/CSV for inventory adjustments and audit data & history; wants concise summaries, simple, easy, and quick to digest audit reports, and clear action cues.
- **Vendor Rep (optional) – External:** Similar needs to field associates.

Satisfaction importance between primary users and recipients is symbiotic—if field associates can conduct quick and efficient inventory audits, store leadership can maintain accurate on-hand

inventory, resulting in a reduction of lost sales and improved customer satisfaction; in turn, store leadership is satisfied with field associates, potentially leading to better sales opportunities.

## **2.4 Operating Environment**

- **Platform:** Android smartphones on SDK 24+.
- **Connectivity:** All core capture works offline; network required only while sending reports.
- **Conditions:** Variable lighting; app provides a flashlight toggle for scanning.

## **2.5 Design and Implementation Constraints**

- User-granted camera and storage permissions are required.
- All data remains on-device in v0.1.0 (no cloud storage/sync, no PII required).
- PDF generation must complete within practical device limits; large images shall be resized as needed.
- Compliance with store policies regarding photography and communications.
- Time constraints (8 weeks total; 4 weeks coding & implementation) may result in a scaled-back finished product (e.g., no dashboard or CSV export option).
- Programming standards follow the MVVM architecture for separation of concerns, promoting reusability and testability of app components.
- The software delivered will be maintained by me (Charles Clark) throughout the application's lifecycle.

## **2.6 User Documentation**

- In-app quick guide ("How to run a shelf audit").
- GitHub README with screenshots and a short walkthrough for online help.
- Sample PDF and brief troubleshooting notes (e.g., scanning in low light, permissions).

## **2.7 Assumptions and Dependencies**

- Field associates/primary users are permitted to photograph shelves and email reports to store leadership.
- A minimal SKU list exists, or barcodes can be scanned on products/tags.

- Email recipients and store/section names are known or configurable.
- **Third-party components:** CameraX, ML Kit, Room, Android Sharesheet, Android PdfDocument or HTML-to-PDF via WebView (fallback)—availability and compatibility assumed on supported devices.

## 3. System Features

The features outlined below are organized by major user-visible services. Each feature includes a short description, priority, typical stimulus/response, and numbered requirements. Priority component ratings for benefit, penalty, cost, and risk are included (each rated on a relative scale from a low of 1 to a high of 9).

### 3.1 Scan or Search Item

#### 3.1.1 Description and Priority

Identify products by barcode scan or manual search.

**Priority: High (P1)**

Benefit 9 / Penalty 8 / Cost 5 / Risk 4

#### 3.1.2 Stimulus/Response Sequences

Stimulus	Response
<b>User opens Scan</b>	Camera view opens with torch toggle and live scan. On detection, item details appear
<b>Barcode unreadable</b>	App suggests manual search instead
<b>Same UPC scanned twice in same audit</b>	Prompt to increment existing item, replace, or ignore
<b>User enters manual Search</b>	Matching product name/UPC is displayed
<b>User clicks on SKU</b>	Item details associated with searched SKU appear

#### 3.1.3 Functional Requirements

- **FR-3.1-1** The app **shall** scan UPC/EAN/GTIN and display a matching item within **5 seconds** under normal lighting (P1).
- **FR-3.1-2** The app **shall** provide manual search and **may** provide type-ahead by name/UPC (P1).
- **FR-3.1-3** The app **should** warn on duplicate scans in the active audit and offer merge/ignore (P2).
- **FR-3.1-4** The app **shall** allow toggling flashlight within the scan screen (P2).
- **FR-3.1-5** On invalid/unknown barcode, the app **shall** show a non-blocking message (such as a snackbar) and offer manual entry (P1).

## 3.2 Record Counts, Notes, and Photos

### 3.2.1 Description and Priority

Capture on-shelf counts with optional notes and photo evidence.

**Priority: High (P1)**

Benefit 9 / Penalty 8 / Cost 5 / Risk 5

### 3.2.2 Stimulus/Response Sequences

Stimulus	Response
User selects item in the audit list	Item card expands to show count field, note, and photo button
User enters <i>count</i> = 0	App suggests adding a photo before report creation
App is interrupted (lock screen/incoming call)	Draft & application state is preserved and restored on return

### 3.2.3 Functional Requirements

- **FR-3.2-1** The app **shall** allow entering integer counts  $\geq 0$  (P1).
- **FR-3.2-2** The app **shall** allow adding an optional free-text note (P1).
- **FR-3.2-3** The app **should** capture at least one photo per item from the camera; gallery attach is optional (P2).

- **FR-3.2-4** When *count* = 0 and no photo is attached, the app **should** suggest attaching a photo (P2).
- **FR-3.2-5** The app **shall** persist edits locally and recover them after process recreation (P1).
- **FR-3.2-6** The app **should** compress photos to a size that balances legibility and PDF size (target  $\leq 400$  KB per embedded image) (P2).

### 3.3 Generate PDF Report

#### 3.3.1 Description and Priority

Generate a timestamped PDF summarizing an audit with lists SKUs and associated recorded counts.

**Priority: High (P1)**

Benefit 9 / Penalty 8 / Cost 6 / Risk 5

#### 3.3.2 Stimulus/Response Sequences

Stimulus	Response
<b>User taps Generate Report</b>	App validates required fields, builds a cover page (store, section, date/time, auditor), and item pages with counts/notes/photos, then shows success or error
<b>Missing required info (e.g., no items)</b>	App explains what's missing and prevents generation

#### 3.3.3 Functional Requirements

- **FR-3.3-1** The app **shall** generate a PDF including store, section (if set), date/time, auditor (if set), and total items (P1).
- **FR-3.3-2** The PDF **shall** list each item (e.g., SKU) with name, UPC, count, notes, and embedded photos (P1).
- **FR-3.3-3** The app **shall** save the PDF to app-private storage and display its location or open option (P1).
- **FR-3.3-4** The app **should** highlight zero/low counts in a “Findings” summary (P2).

- **FR-3.3-5** A 10-item audit with 10 photos (if set) **shall** complete generation in  $\leq 30$  seconds on a mid-range device (P1).
- **FR-3.3-6** If generation fails, the app **shall** show a clear error and allow retry without data loss (P1).

## 3.4 Email/Share Report

### 3.4.1 Description and Priority

Send the generated PDF report via Android Sharesheet or an email app.

**Priority: High (P1)**

Benefit 8 / Penalty 7 / Cost 3 / Risk 3

### 3.4.2 Stimulus/Response Sequences

Stimulus	Response
<b>User taps Share Report</b>	Sharesheet opens with the PDF attached and a prefilled subject line
<b>No email app installed</b>	App displays a helpful message and suggests other share options (e.g., text message)

### 3.4.3 Functional Requirements

- **FR-3.4-1** The app **shall** invoke the Sharesheet with the PDF attached as application/pdf (P1).
- **FR-3.4-1** The app **shall** prefill the subject line with store, section, and date (P1).
- **FR-3.4-1** The app **should** prefill recipients from settings when available (P2).
- **FR-3.4-1** The app **shall** never send automatically; user confirmation is required (P1).

## 3.5 History and Export

### 3.5.1 Description and Priority

Browse past audits, open their PDFs, and export CSV.

**Priority: Medium (P2)**

Benefit 7 / Penalty 4 / Cost 4 / Risk 3

### 3.5.2 Stimulus/Response Sequences

Stimulus	Response
<b>User taps History</b>	List of card(s) containing audits appear with date, store, section (if set); filters available
<b>User selects Export CSV</b>	App generates a CSV and opens Sharesheet for the file

### 3.5.3 Functional Requirements

- **FR-3.5-1** The app **shall** list past audits with date, store, section, and quick actions (open/share/export) (P2).
- **FR-3.5-1** The app **should** filter by date range, store, or section (P2).
- **FR-3.5-1** The app **may** export selected audit(s) to CSV with headers: Date, Store, Section, Item, UPC, Count, Notes (P2).
- **FR-3.5-1** CSV generation errors **shall** surface clearly and permit retry (P2).

## 3.6 Add SKU

### 3.6.1 Description and Priority

Add SKU to catalog with associated details through manual entry or scan help (for UPC).

**Priority: High (P1)**

Benefit 9 / Penalty 9 / Cost 4 / Risk 5

### 3.6.2 Stimulus/Response Sequences

Stimulus	Response
<b>User taps Add SKU</b>	Empty SKU fields are displayed for a single addition
<b>User taps Scan</b>	App opens barcodescanner and fills in associated SKU details if UPC is scanned
<b>User taps Add</b>	App enter SKU into SKU catalog for audits

### 3.6.3 Functional Requirements

- **FR-3.6-1** The app **shall** allow user to add SKU to catalog (P1).
- **FR-3.6-2** The app shall display SKU fields for addition (SKU, UPC, brand, size) (P1).
- **FR-3.6-3** The app shall enter SKU into catalog list for audits (P1).

## 4. External Interface Requirements

### 4.1 User Interfaces

- **Visual style:** Simple, high-contrast (accessibility-focused) design using Material Design 3; consistent primary actions at the bottom (e.g., Generate PDF, Share Report); side navigation drawer for easy navigation between screens.
- **Standards:** Large touch targets and meaningful content descriptions for accessibility; consistent error toasts/snackbars.
- **Screens:** Home, Scan/Search, Audit (item cards with count/note/photo), Report Preview/Confirm, History.
- **Error messages:** Plain English; include next step if possible (e.g., “Try manual search”).
- **Help:** One-page “How to run a shelf audit.”

### 4.2 Hardware Interfaces

Single rear **camera** required for scanning/photos; app-private **storage** for photos and PDFs; optional access to gallery (deferred); each software interface will support **Android SDK 24+** devices.

### 4.3 Software Interfaces

The connections between this product, MerchTools, and other specific software components are outlined below:

- **Operating system (target):** Android SDK 24+.
- **Tools & libraries:** CameraX; ML Kit for barcode scanning; Room for local database; Android Sharesheets/Intents; Android PdfDocument or HTML-to-PDF via WebView print framework (fallback).

- **Data input:** SKU list for store; UPC/GTIN data from scanning; stored images (optional).
- **Data outputs:** PDF, CSV via Sharesheet.

*\*No server APIs in v0.1.0\**

## 4.4 Communications Interfaces

The primary communications functions required by MerchTools will be associated with **Email/Share** using Android Sharesheet; this relies on user-installed apps (e.g., Gmail) that supply their own encryption methods and communication/network protocols.

**File MIME types** associated with this product for the purposes of identification and sharing/serving are application/pdf, text/csv.

No background uploads will occur, and sharing occurs only after explicit user action.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

Performance requirements for system feature [3.1 Scan or Search Item](#):

- **NFR-1.1** Scan detect & identify in  $\leq 2s$  in typical lighting on a mid-range device.

Performance requirements for system feature [3.3 Generate PDF Report](#):

- **NFR-1.2** PDF generation for 10 items with 10 photos in  $\leq 30s$ .

Performance requirements for system feature [3.5 History and Export](#):

- **NFR-1.3** History list loads in  $\leq 10s$  for 100 audits.

### 5.2 Safety Requirements

- **NFR-2.1** Camera use **shall** halt when the app is backgrounded.

### 5.3 Security Requirements

- **NFR-3.1** All data (photos, PDFs, audit records) **shall** remain on device by default in v0.1.0.
- **NFR-3.2** The app **shall** not collect PII unless the user opts to set an auditor name or email recipients.

- **NFR-3.3** The app **shall** not allow data collection (i.e., auditor name and email/share recipients) before explicit permission is asked and granted by the user.
- **NFR-3.4** Shared files **shall** be created in app-private space and exposed via content URIs with read-only flags.
- **NFR-3.5** The app **shall** provide a privacy policy (for any data that may be collected) that is clear and accessible; clearly states the purpose and data types; explains data use and sharing; details security measures; informs users of their rights; obtain explicit consent for data collection; comply with any legal frameworks and app store policies.

## 5.4 Software Quality Attributes

- **Usability:** Simple and efficient primary workflow with “How to” guide enables ease of use and learning.
- **Reliability:** Autosave protects against process death; application state is preserved when backgrounded; no data loss on rotation or interruption.
- **Maintainability:** MVVM architecture enables separation of concerns with clear interfaces.
- **Reusability:** MVVM architecture can provide scaffolding such that components (e.g., ViewModels) can be reused in other applications.
- **Testability:** MVVM architecture allows efficient testing; ViewModels and use-cases unit-testable with fakes; Room DAOs instrumented tests; Compose UI tests.
- **Accessibility:** All actionable controls have labels and large touch targets; minimum contrast per WCAG AA where feasible.
- **Portability:** Works on Android SDK 24+ devices in portrait orientation; potential for cross-platform (e.g., iOS) portability in the future with Kotlin Multiplatform.

## 6. Other Requirements

1. **Policy/Legal:** Comply with store policies on photography and reporting; no automatic transmission/sharing.
2. **Internationalization:** English-only v0.1.0; future support for other languages planned thereafter.

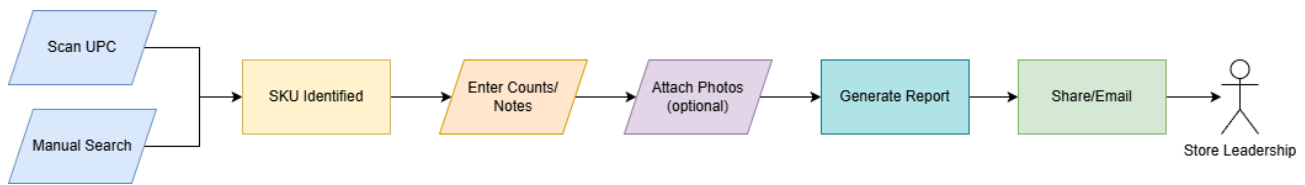
3. **Data Retention:** PDFs and photos retained until the user deletes them or uninstalls the app; no auto-purge in v0.1.0.
4. **File Size Targets:** Default photo compression aims for PDFs  $\leq 8$  MB for a 10-item audit with 10 photos.
5. **Branding:** Minimal; store/vendor/developer branding may be added in a later release.

## Appendix A: Glossary

- i. **Audit:** A short review of items on a shelf/section with counts and photos.
- ii. **UPC:** Universal Product Code for identifying products.
- iii. **EAN:** European Article Number for identifying products (compatible with UPCs).
- iv. **GTIN:** Global Trade Item Number found under the barcode on a product's packaging.
- v. **SKU:** Stock Keeping Unit (specific product).
- vi. **OOS:** Out of Stock.
- vii. **MVVM:** Model-View-ViewModel; software architecture that separates an application into three interconnected components to enhance maintainability, testability, and organization.
- viii. **Compose/Jetpack Compose:** Android's recommended modern toolkit for building native UI; simplifies and accelerates UI development with less code, powerful tools, and intuitive Kotlin APIs.

## Appendix B: Analysis Models

**High-level data diagram:**



**User Flow:** Scan/Search → Add count/note/photo → Repeat for items → Generate PDF → Share → History.

**Data Entities:** Store, Section, Audit, AuditItem, Photo, Sku.

Observation notes and interview summaries will be added here after initial field sessions.

## Appendix C: Issues List

Status	Details	Initial Date	Revised Date
Open	Decide minimum photo resolution for legible evidence without large PDF size	11/2/2025	
Open	Confirm default report recipients per store	11/2/2025	
Deferred	Gallery photo attach; dashboard; CSV	11/2/2025	
Risk	Low-light scanning reliability on older devices	11/2/2025	