Software Functional Requirements Document for Smart Checkout Nexus

Alex M. Frear

College of Science, Engineering, and Technology, Grand Canyon University

Course Number: CST-326 Written and Verbal Communication for Software Development

Professor Stephanie Macuga

04/5/2025

Table of Contents

1. Introduction

1.1 Purpose of the Document

The purpose of this Functional Specification Document (FSD) is to define the functional and non-functional requirements for the *Smart Checkout Nexus* project. This document provides a detailed breakdown of system behaviors and specifications derived from the initial user stories. It serves as a roadmap for development and testing, ensuring alignment with the simulated client's needs and expectations.

1.2 Project Scope

Smart Checkout Nexus is a simulated retail station solution that integrates real-time database functionality with accounting and inventory systems. The software aims to provide a fast, intuitive checkout experience with support for hardware devices such as scan guns, scales, and handheld attendant devices. The scope includes core front-end and back-end functionality, hardware interaction, and accessibility features, excluding third-party system integrations beyond SAP and MySQL.

1.3 Related Documents

- Milestone 1 – Team Charter
- Milestone 1 – User Stories Document
- CST-326 Agile Board (Trello Screenshot)

1.4 Terms/Acronyms and Definitions

Term/Acronym	Definition
FSD	Functional Specification Document
GUI	Graphical User Interface
SAP	Systems, Applications, and Products in Data Processing (Enterprise Resource Planning software)
SQL	Structured Query Language
UX	User Experience
LAN	Local Area Network

1.5 Risks and Assumptions

- The project is completed by a solo developer, which may pose time and resource constraints.
- Hardware interactions (e.g., scan gun, Bluetooth devices) are simulated.
- SAP and MySQL system connections are mocked for prototype purposes.

- Accessibility requirements will be designed according to standard WCAG (Web Content Accessibility Guidelines).
- Diagrams and wireframes will be created based on best practices but may not reflect a final production-ready UI.

---

2. System/Solutions Overview

The *Smart Checkout Nexus* is a web-based software solution designed to improve efficiency and accessibility at a retail self-checkout station. It integrates a real-time database system with SAP and MySQL for inventory and sales synchronization and supports both customer and attendant interactions via multiple hardware components.

The solution includes:

- A customer-facing interface for scanning, loyalty rewards, receipts, and accessibility
- An attendant-facing handheld interface with keyboard input, overrides, and remote access
- Real-time updates to backend systems and performance metrics such as quick on-screen updates
- Support for devices such as LAN connections, Bluetooth, barcode scan guns, and product scales

The main goals of the system are:

- Improve customer checkout experience and flow
- Reduce manual errors in pricing and promotions
- Enable attendants to assist users remotely via mobile devices
- Maintain up-to-date records of transactions and inventory

2.1 Context Diagram / Interface Diagram / Data Flow Diagram

Data Flow Summary:

1. Customer scans item → System updates on-screen item list in real time.
2. Optional loyalty card scan or manual entry → Pulls customer profile and applies rewards.
3. Scale interaction → Weight-based item data is calculated and displayed.
4. Coupon entry (triggers attendant approval) → Discount applied on approval.
5. Transaction submitted → Updates sent to SAP & MySQL for inventory and accounting.
6. Receipt generated → Option to print or email receipt.

7. Attendant device access → Allows remote override, product lookup, and assistance.

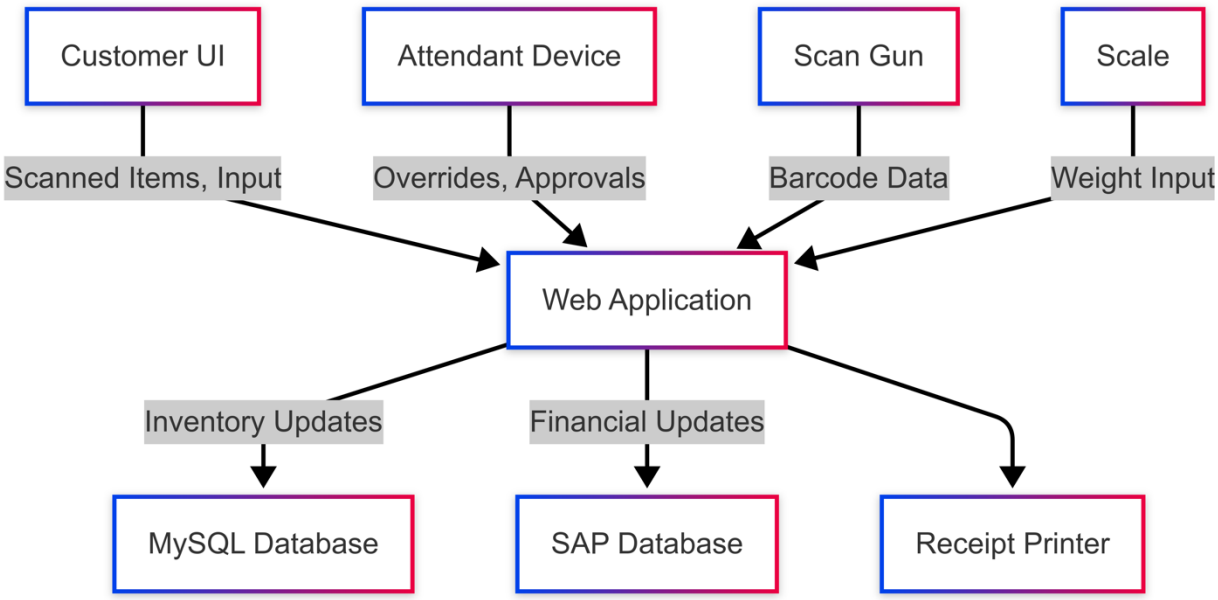

Figure 1 System Architecture Diagram for Smart Checkout Nexus. This diagram illustrates the interaction between user interfaces, hardware devices, the central web application, and the integrated backend databases.

---

3. Functional Specifications

3.1 Architecture

The *Smart Checkout Nexus* architecture follows a modular web application structure, using a client-server model with real-time database integration. The system is designed to simulate a self-checkout environment commonly used in retail settings. The architecture includes the following components:

Frontend (Client Layer)

Built as a responsive web interface for customer-facing and attendant-facing devices. Includes:

- Touchscreen interface (customer station)
- Handheld device interface (attendant use)
- Real-time updates via API calls

Backend (Server Layer)

Handles application logic, request routing, authentication, and interactions with the database. Responsible for:

- Processing scanned items and loyalty card data
- Managing promotional rules and restrictions (e.g., coupon handling, restricted items)
- Enabling manager overrides and attendant controls

Database Layer

Real-time integration with:

- **SAP** for inventory and financial data
- **MySQL** for customer profiles, transactions, and promotions

Hardware Integration Layer

Interfaces with devices including:

- Barcode scan gun
- Item scale
- Receipt printer
- Bluetooth or LAN-connected peripherals

3.2 Purpose/Description

The purpose of this section is to provide detailed specifications of the *Smart Checkout Nexus* system, ensuring traceability from user stories to implementation. Each subsection outlines the components, behaviors, and interactions required for development. These specifications are intended for use by developers, testers, and stakeholders to validate the system's design and expected performance.

3.3 Sitemap

The system includes two primary user paths: one for customers and one for attendants. The sitemap below outlines the structure of the web application:

Customer Interface Sitemap

- Home Screen
  - Scan Item
  - Enter Loyalty Info
  - View Cart

    - Apply Coupon
    - Call for Assistance
    - Proceed to Checkout
        - Select Receipt Option (Print or Email)

Attendant Interface Sitemap

- Login
    - Dashboard
        - Respond to Assistance Call
        - View Active Stations
        - Perform Override
        - Product Lookup
        - Upload Promotions

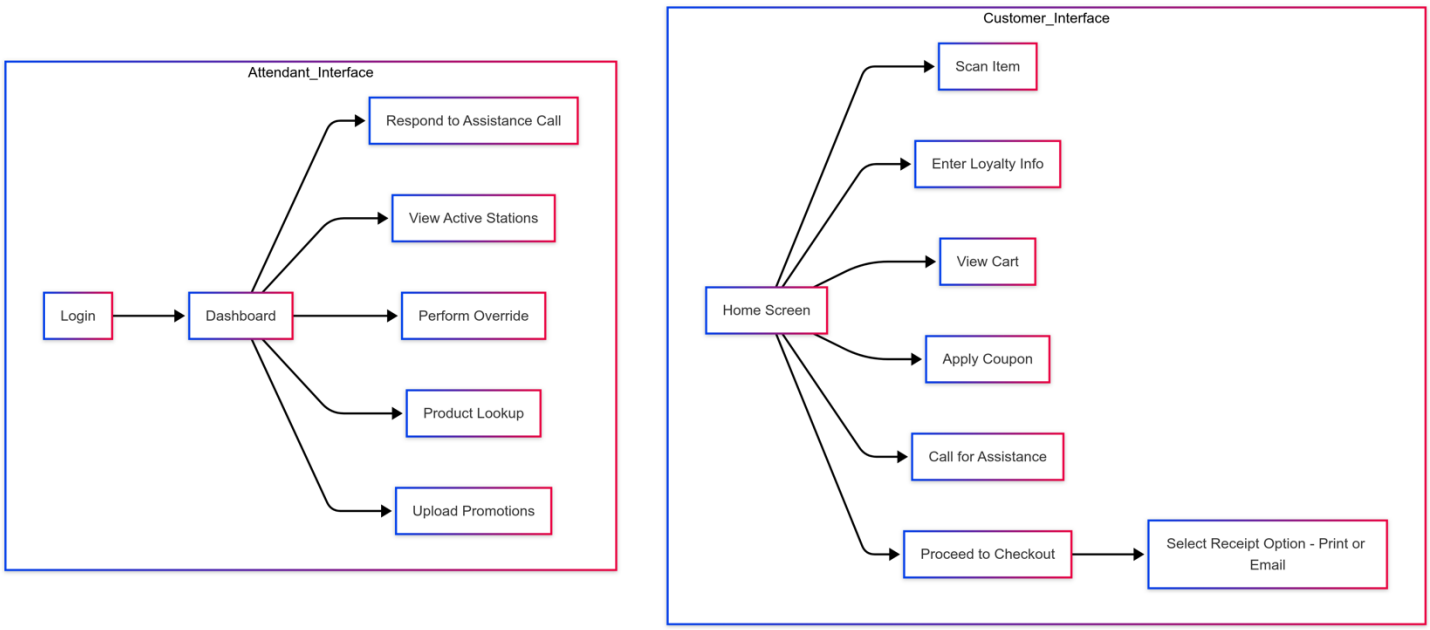

Figure 2 Sitemap Diagram for Smart Checkout Nexus Sitemap Diagram for Smart Checkout Nexus. This visual illustrates the primary navigation flow for both the Customer and Attendant Interfaces, including an optional checkout path without coupon entry.

3.4 Wireframe

Customer Interface – Home Screen

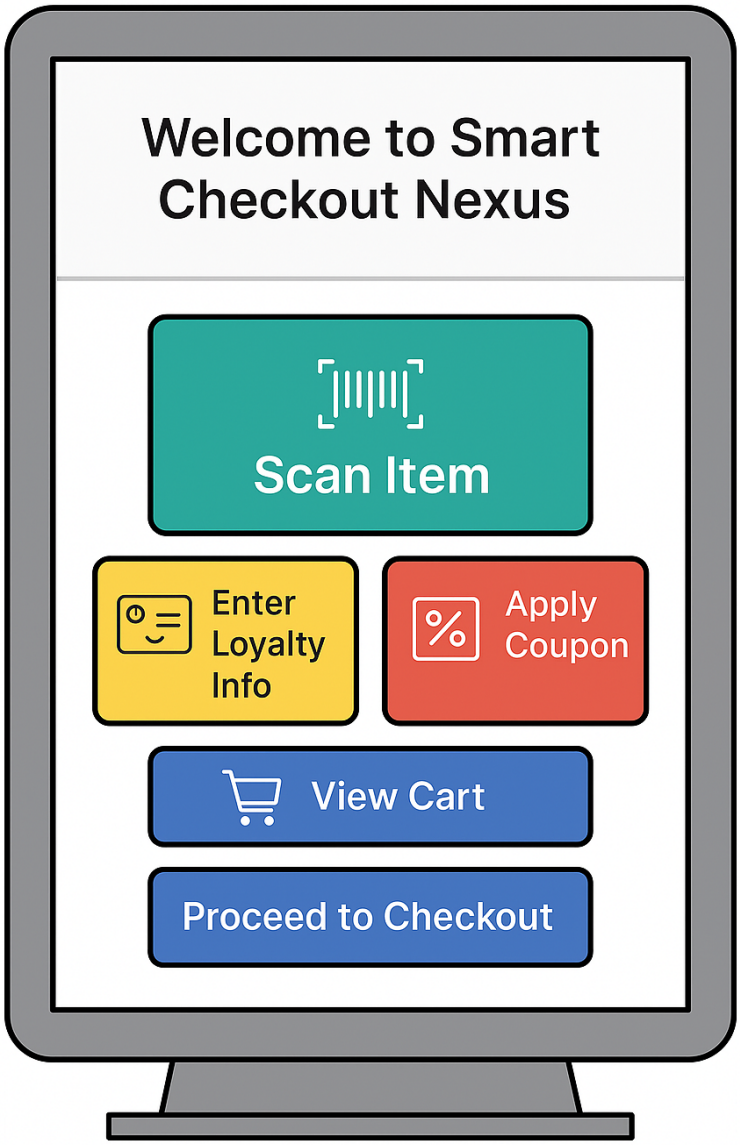

Figure 3 A self-checkout touchscreen interface displaying primary user options. Features include Scan Item, Enter Loyalty Info, Apply Coupon, Call for Assistance, View Cart, and Proceed to Checkout.

Customer Interface – Produce Lookup Screen:

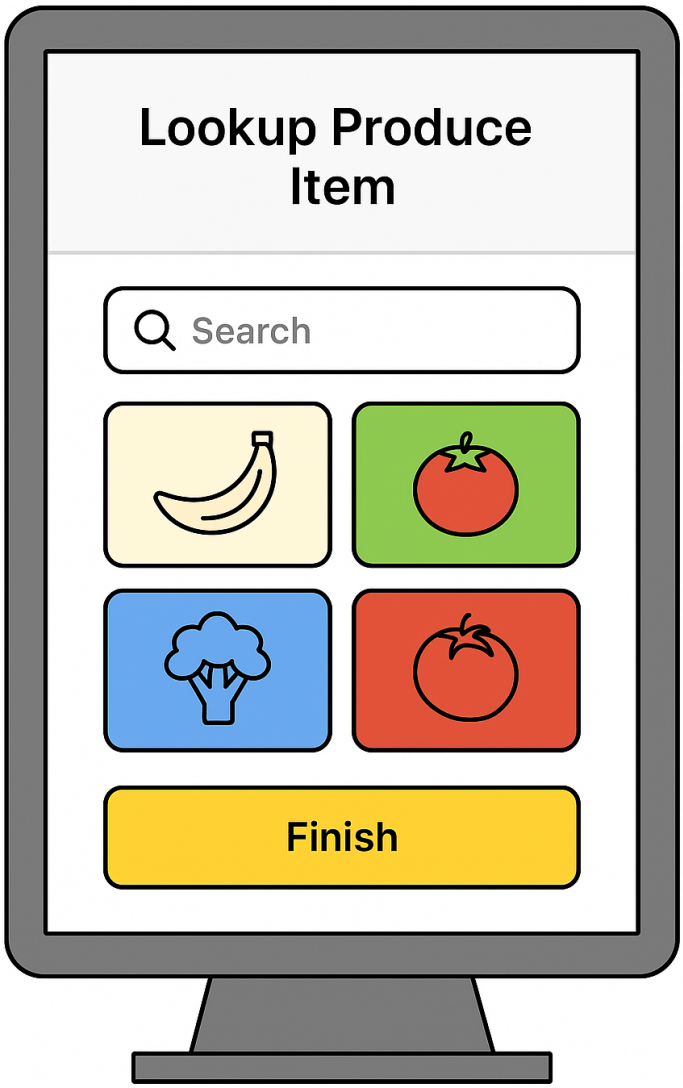

Figure 4 This screen allows customers to locate produce items by typing a name or selecting from visual image tiles. This feature supports R_15 by streamlining produce entry into the cart when barcode scanning is unavailable.

Customer Interface – Scale Interaction Screen:

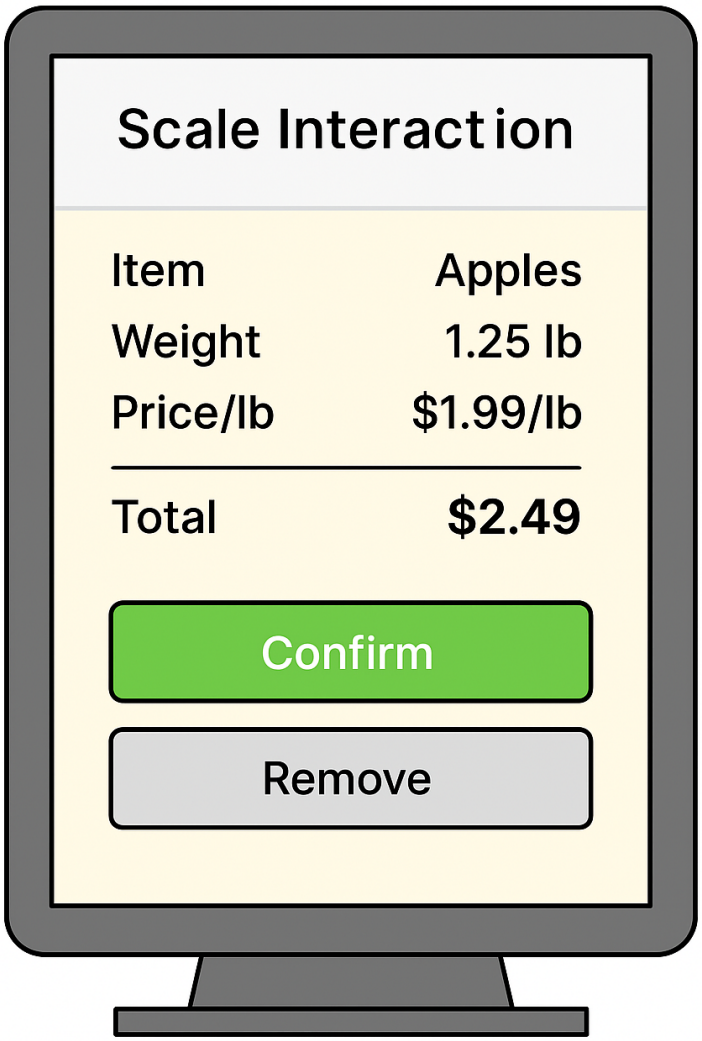

Figure 5 This screen appears when an item requiring weight input is scanned. The display shows the item name, measured weight, calculated price, and includes a visual confirmation for customer verification.

Customer Interface – Cart View

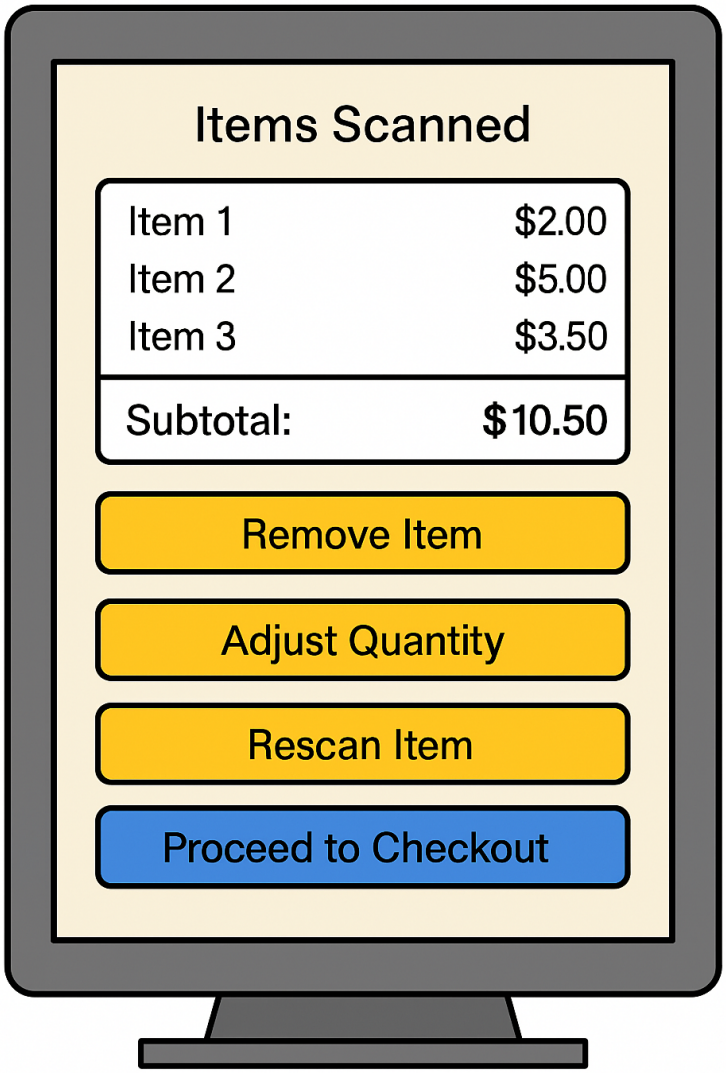

Figure 6 A screen summarizing scanned items with their prices and subtotal. Includes action buttons for Remove Item, Adjust Quantity, Rescan Item, and Proceed to Checkout.

Customer Interface – Receipt Option

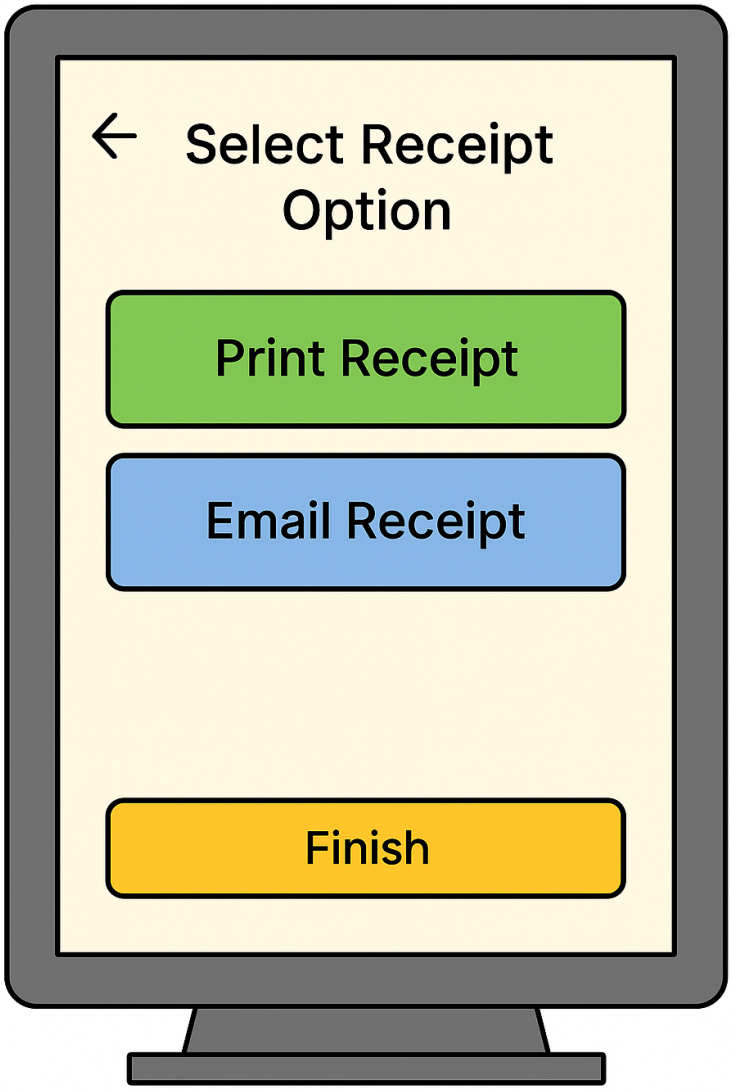

Figure 7 Final step in the customer flow. Allows the user to select their preferred receipt delivery method (Print or Email) and complete the transaction with a Finish button. A back arrow allows returning to the cart view.

Attendant Interface – Login

Figure 8 A mobile-friendly login screen for attendants using a rugged handheld device. Features input fields for Username and Password, and a Login button.

Attendant Interface – Dashboard

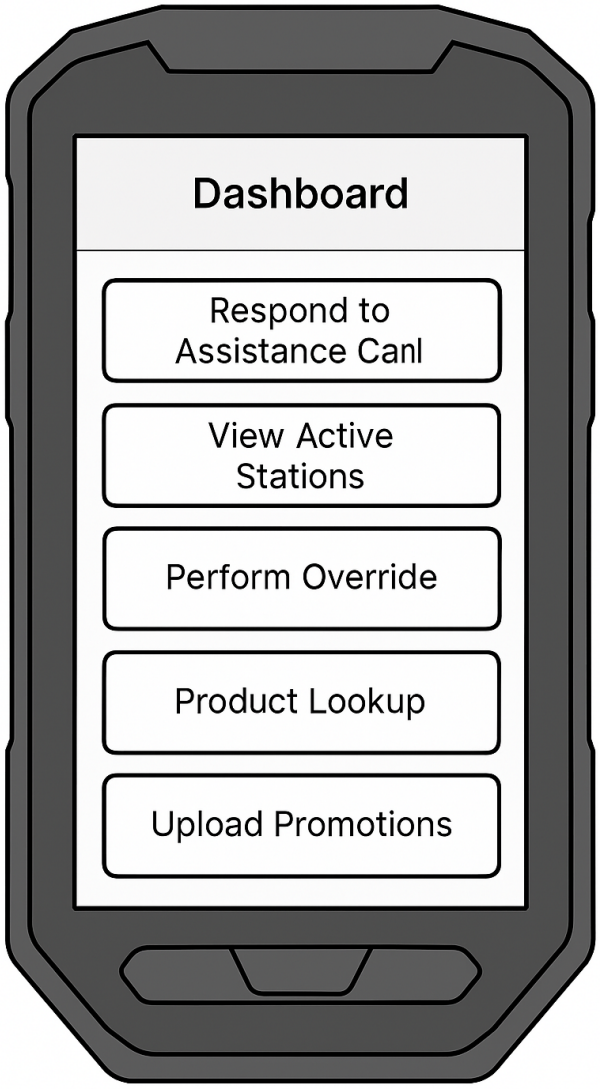

Figure 9 *The main control panel for store attendants. Includes quick access buttons for Responding to Assistance Calls, Viewing Active Stations, Performing Overrides, Looking Up Products, and Uploading Promotions.*

Attendant Interface – Override Request Pop-Up:

Figure 10 This pop-up appears on the attendant's handheld device when a restricted item or coupon requires approval. The attendant can approve or deny the request and add an optional comment for logging. Supports R_7, R_11, and NFR_3.

Attendant Interface – Respond to Assistance Call Screen:

Figure 11 This screen displays incoming help requests from customer kiosks. Attendants can acknowledge, respond, and mark requests as resolved. Supports R_8 (Call for Assistance) and enhances real-time staff responsiveness.

3.5 Functional Requirements

Use Case ID: UC-001 – Hardware and Connectivity

ID	Functional Requirement
FR-1.1	The system shall support LAN connectivity for stable data transmission to backend systems.
FR-1.2	The system shall support Bluetooth connectivity to interface with wireless peripherals such as handheld devices and scales.
FR-1.3	The station shall connect to a scan gun to capture barcode inputs for large or oversized items.
FR-1.4	The system shall read weight input from a connected digital scale and apply pricing based on configured per-unit weight rates.

Use Case ID: UC-002 – Attendant and Manager Tools

ID	Functional Requirement
FR-2.1	The attendant shall be able to access the system via a handheld device connected over Bluetooth or LAN.
FR-2.2	The attendant handheld device shall include a QWERTY keyboard and number pad interface.
FR-2.3	The system shall allow attendants to approve scanned coupons before they are applied.
FR-2.4	The system shall allow manager overrides to be executed via both the station and the attendant device.
FR-2.5	The attendant dashboard shall allow for uploading of daily specials and weekly ad pricing.
FR-2.6	The attendant shall be able to respond to customer assistance requests in real-time.

Use Case ID: UC-003 – Customer Interaction

ID	Functional Requirement
FR-3.1	The customer shall be able to scan a loyalty card or manually enter phone/email to retrieve rewards data.
FR-3.2	The customer shall have the option to print or email a receipt upon checkout.
FR-3.3	The system shall allow the customer to skip bagging an item when prompted.
FR-3.4	The customer shall be able to look up produce items by name or image selection.
FR-3.5	The customer shall have the ability to call for attendant assistance through an on-screen button.

Use Case ID: UC-004 – Database Integration

ID	Functional Requirement
FR-4.1	The system shall update inventory and accounting data in real time using SAP integration.
FR-4.2	The system shall store transaction data and user profiles in a MySQL database.

FR-4.3	The system shall synchronize loyalty rewards and promotions data across both SAP and MySQL databases.

3.6 Field-Level Specifications

Form Elements

Call-Out	Field Label	UI Control	Mand?	Editable	Data Type	Value Set	Default Value	Data Example	Data Source
1	Loyalty Input	Textbox	No	Yes	Alphanumeric	N/A	N/A	alex@email.com	User Entry
2	Coupon Code	Textbox	No	Yes	Alphanumeric	N/A	N/A	SAVE20	User Entry
3	Produce Lookup	Search Box	No	Yes	Text/Image	Fruit, Veggies, etc.	N/A	Bananas	Local DB
4	Weight Input	Sensor	Yes	No	Decimal	N/A	N/A	1.45	Scale Device
5	Email Receipt Field	Textbox	No	Yes	Email	N/A	N/A	receipt@email.com	User Entry

Form Business Rules and Dependencies

Field Label	Validation / Business Rules	Error Messages	Data Dependencies	Additional Info
Loyalty Input	Must match phone/email pattern if manually entered	“Please enter a valid phone number or email.”	None	Optional step before checkout
Coupon Code	Must match active promo code in DB	“Invalid or expired coupon code.”	SAP/MySQL	Requires attendant approval
Produce Lookup	Input must match available product list	“Item not found. Try another keyword.”	Local DB	Includes autocomplete suggestions
Email Receipt Field	Must be a valid email format	“Invalid email format.”	None	Optional if user selects email option
Weight Input	Value must be > 0	“Invalid weight. Please rescan item.”	Device sensor input	Used to determine pricing

Buttons, Links, and Icons

Button/Link/Icon Label	OnClick Event	Other Event	Visible	Enabled vs Disabled	Navigate To	Validation	Dependencies
Scan Item	Starts item scan and triggers	N/A	Yes	Enabled	Item Display	None	Requires Scan Gun Connected

	database lookup						
Apply Coupon	Sends coupon code to DB and requests attendant approval	OnMouseHover: show tool tip	Yes	Enabled if coupon code entered	Same page	Must match active coupon	Coupon Code field
Call Attendant	Notifies attendant system and highlights station location	N/A	Yes	Always enabled	N/A	None	None
Skip Bagging	Marks item as not bagged, skips weight confirmation	N/A	Yes	Enabled after item scanned	Next item scan	None	Item must be scanned first
Finish & Checkout	Finalizes order and shows receipt options	OnMouseHover: show tip	Yes	Enabled after all items scanned	Receipt Options	Validates all required fields (e.g., weight)	All item scans complete

3.7 Nonfunctional Requirements

ID	Nonfunctional Requirement
NFR-1	The physical dimensions of the self-checkout station shall not exceed 4 feet in height, 3 feet in width, and 6 feet in length to comply with facility space constraints.
NFR-2	The customer interface shall use a touchscreen input method that is intuitive and responsive for users of all ages.
NFR-3	The system shall require attendant approval before restricted items such as alcohol, tobacco, and cold medicine can be purchased.
NFR-4	The station's touchscreen and payment input areas shall be fully accessible to users in a standard wheelchair, including screen height, angle, and reach distance.
NFR-5	After each item is scanned, the system shall update the on-screen itemized list within 15 seconds to maintain a smooth user experience.

4. References

The following documents and tools were used as inputs for developing this Software Functional Requirements Document:

- Milestone 1 – Team Charter

- Milestone 1 – User Stories Document
- CST-326 Agile Board (Trello)
- CST-326-RS-T3 Software Functional Requirements Document Template (Provided)
- EARS – Easy Approach to Requirements Syntax (Course Resource)
- SAP and MySQL documentation (simulated for project context)
- WCAG 2.1 Guidelines for accessibility reference

---

5. Open Issues

Issue ID	Issue Description	Raised By	Raised On	Solution/Decision	Resolved By	Resolved On	Status
001	Final decision on screen layout and wireframe design	Alex	03/29/2025	All wireframes created and added to Section 3.4 of the document	Alex	04/05/2025	Resolved
002	Diagram creation for system architecture and data flow	Alex	03/29/2025	Context diagram finalized and added to Section 2.1	Alex	04/05/2025	Resolved
003	Testing method for Bluetooth device simulation	Alex	03/29/2025	Pending future milestone (testing not part of this deliverable)	—	—	Open

Note: All issues are being tracked and managed within the CST-326 Agile tool (Trello).

---

6. Appendix

This section contains any supplementary materials referenced or used during the development of this document. Items may include:

- Finalized wireframes for both the Customer and Attendant Interfaces (see Section 3.4)
- System architecture and data flow diagrams (see Section 2.1)

- Notes from internal planning and requirement breakdowns
- External documentation references including SAP, MySQL, and WCAG 2.1 guidelines
- Requirement tracking screenshots from Trello (see Figures 1 and 2)
- **Trello Board Access**
  To view the full requirement tracking board with cards, checklists, comments, and wireframes, visit: https://trello.com/b/jbQi5KMt/smart-checkout-nexus-backlog

---

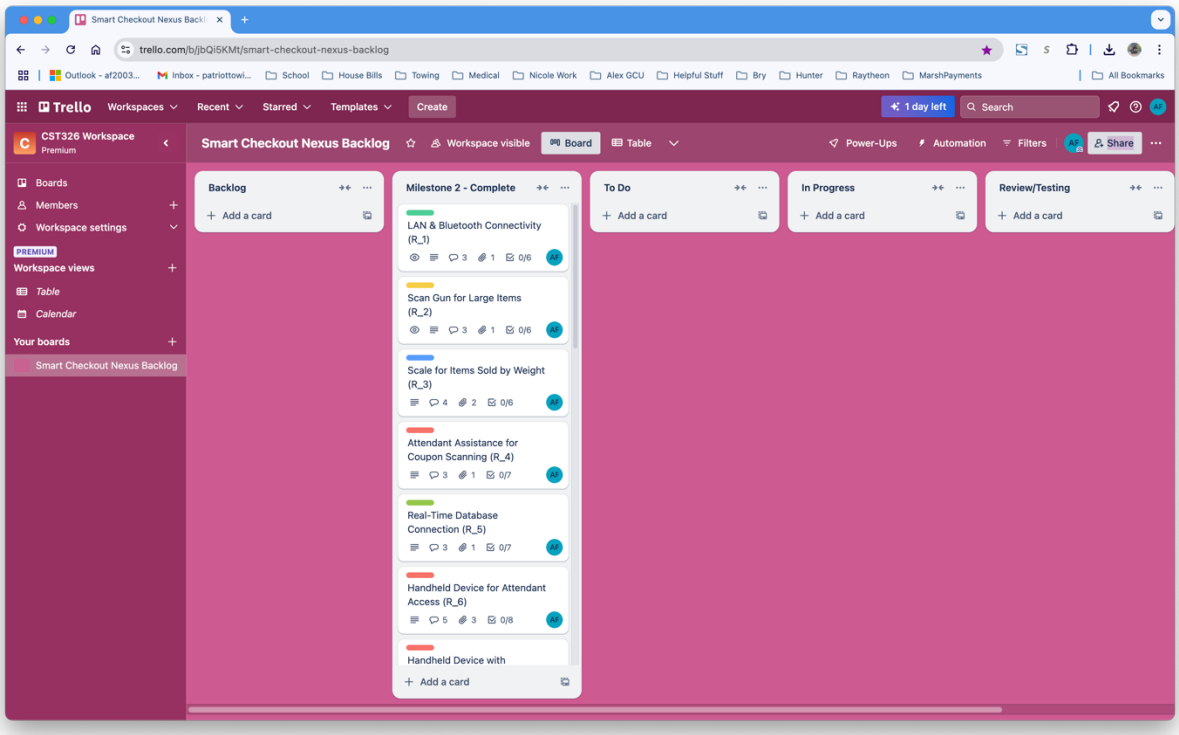

Figure 12 Trello board showing the backlog, status columns, and milestone progress for all user stories and requirements.

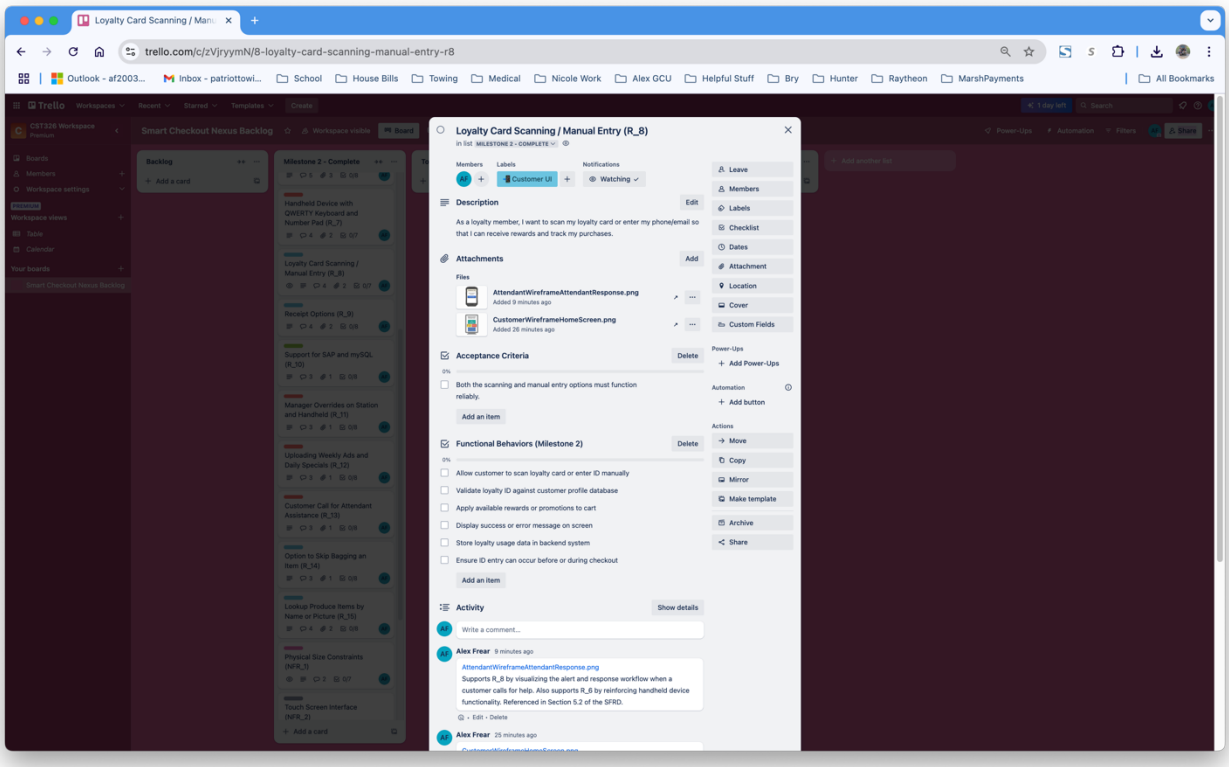

Figure 13 A sample Trello card with completed checklist, summary comment, and wireframe attachment.

Trello was used to manage and track each functional (R_1–R_15) and non-functional requirement (NFR_1–NFR_5). Cards were labeled by feature type, updated with milestone-specific checklists, and moved to the “Milestone 2 – Complete” column once fully documented.