Technical Design 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

04/12/2025

Table of Contents

Introduction

Smart Checkout Nexus is a self-service retail solution designed to revolutionize the checkout experience for both customers and store attendants. The system provides a sleek, intuitive interface for customers to scan and bag items, request assistance, and complete payment—all without requiring constant supervision. An integrated attendant dashboard provides real-time override capabilities, coupon scanning support, and monitoring tools to ensure a secure and efficient flow. With responsive UI elements and smart features, Smart Checkout Nexus enhances usability, reduces wait times, and integrates smoothly into grocery and retail environments. The system supports both credit/debit processing and receipt handling, while also complying with accessibility standards and system security expectations.

Scope

This technical design document outlines the architectural and functional blueprint required to implement Smart Checkout Nexus. It defines the core infrastructure, software components, communication protocols, supported platforms, and database design used to bring the features described in the Requirements Document to life. Key focus areas include the user interface dialogs, background tasks, data management schema, API endpoints, and client-server responsibilities. Additionally, it addresses how each requirement will be realized and tested, ensuring traceability and alignment with Agile methodologies. The document also includes diagrams, wireframes, and flow descriptions that guide the development of all user-related features, integrations, and non-functional compliance requirements.

System Architecture / Infrastructure

The Smart Checkout Nexus system is built using a modular client-server architecture that supports both customer and attendant interfaces. The infrastructure includes front-end and back-end components, connected through secure API calls, and a centralized cloud-hosted database.

The system consists of the following main components:

- **Customer Interface (POS Scanner):** A touch-based self-checkout kiosk that runs in a browser or dedicated app, allowing users to scan items, manage carts, and complete transactions.
- **Attendant Interface:** A separate web-based dashboard for store employees to respond to assistance calls, approve overrides, and monitor active stations.
- **Backend Server:** A RESTful API server built with Node.js and Express, which handles business logic, session tracking, and integration with third-party services like payment processors.
- **Database:** A MySQL database used for operational data and integrated with SAP for inventory and financial updates.
- **Payment Device Integration:** A Verifone payment terminal connected via USB or network interface to process credit and debit card transactions.
- **Cloud Hosting Environment:** The application is deployed using a platform like Vercel or Heroku for scalable frontend/backend hosting, with authentication handled through secure web protocols.

The high-level architecture is depicted in the following diagram:

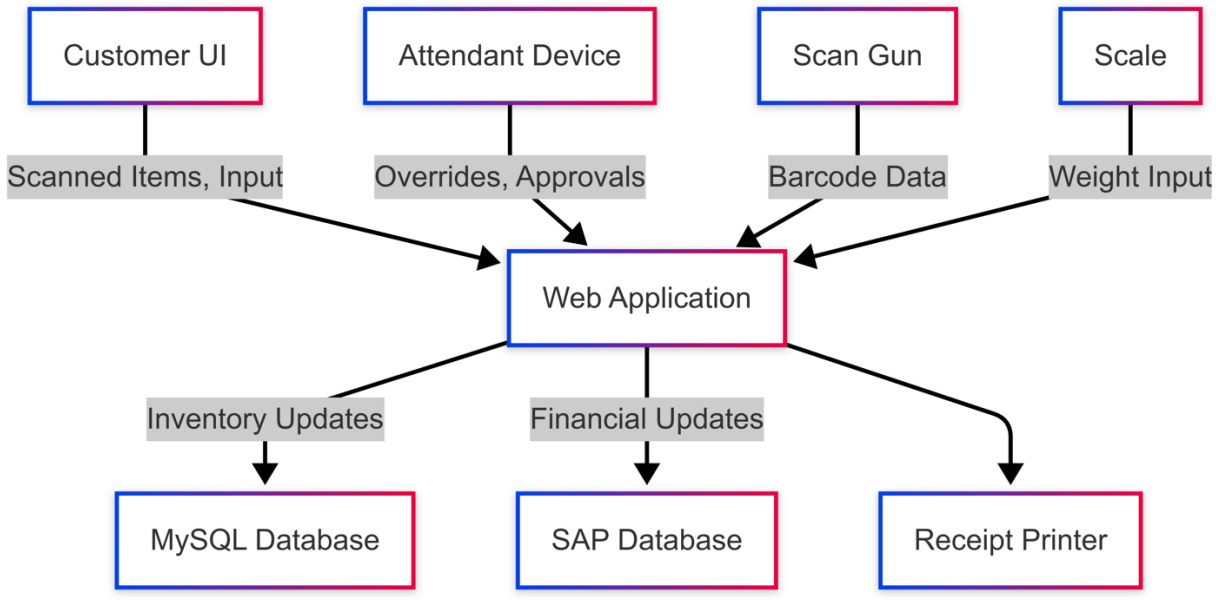

Figure 1 Figure 2 System Architecture Diagram for Smart Checkout Nexus. This diagram illustrates the interaction between user interfaces, hardware devices, the central web application, and the integrated backend databases.

The modularity of this design allows for independent development, testing, and scaling of each component. Features such as call attendant support and override management are handled asynchronously to ensure uninterrupted customer transactions. The system supports integration with existing store infrastructure while remaining flexible for future updates.

User Dialogs and Control Flow

Smart Checkout Nexus presents an intuitive sequence of user dialogs that guide both customers and attendants through various checkout-related interactions. Each screen is designed to support task flow efficiency, user clarity, and accessibility.

The system begins with a **Home Screen** displaying a welcome message and two options: **Begin Checkout** and **Call Attendant**. If a user selects **Call Attendant**, a lock screen appears until the attendant responds via the Attendant Dashboard. If **Begin Checkout** is selected, the user is taken to the **Cart View**, where scanned items are displayed in real-time alongside their prices.

During the checkout process, users interact with the **Produce Lookup** screen to search and select non-barcoded items. If an item requires weighing, the **Scale Interaction** screen prompts users to place the item on the scale, and the web application automatically fetches the weight input.

As scanning progresses, the system continuously updates the cart and total. If a restricted item (such as alcohol or cold medicine) is scanned, the **Override Pop-Up** appears and locks the screen until an attendant verifies the user and approves the transaction. The attendant uses their own **Attendant Login** and **Dashboard** to respond to alerts, perform overrides, and assist with coupon scanning.

Once the user completes scanning, they are directed to the **Receipt Option** screen where they choose between a printed or emailed receipt. At this point, the **Pay UI** is displayed, allowing selection between credit/debit card or cash payment. Credit/debit transactions trigger the Verifone terminal sequence, which is handled externally and updates the financial system upon completion.

The **Respond to Assistance Call** dialog is a specialized screen available only on the attendant interface, providing contextual information on which station sent the request and what type of help is needed (e.g., override, coupon issue, scale discrepancy).

All dialogs are built for touch interaction and screen-reader compatibility, ensuring accessibility compliance. Below is a flowchart representing the general control flow during a typical customer transaction:

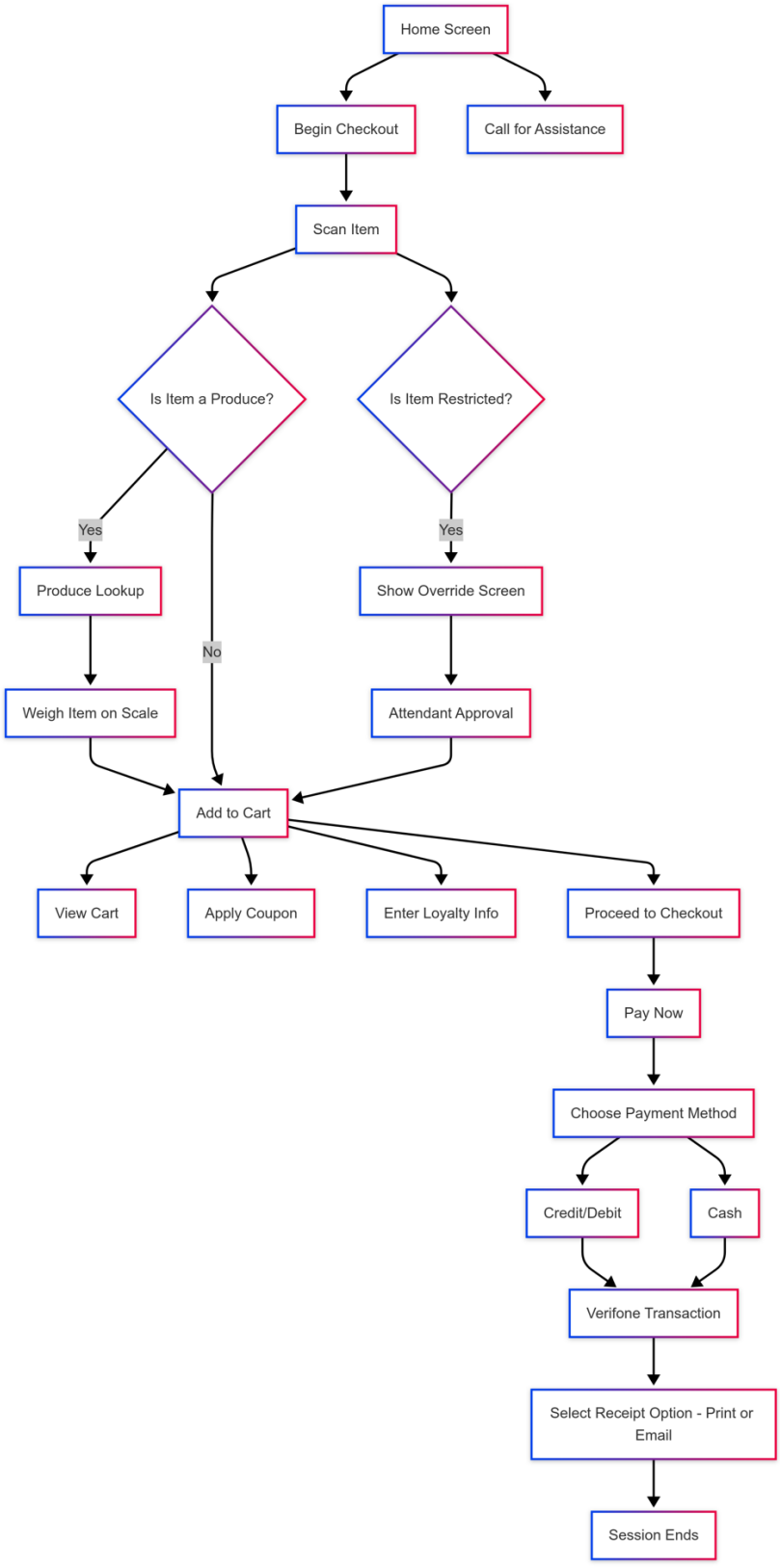

Figure 3 Smart Checkout Nexus – Customer Transaction State Diagram

This control flow ensures that all necessary steps—including assistance calls, payment, and receipt handling—are handled in a clear, step-by-step fashion while allowing the user to backtrack or call for help at any time.

Background Tasks

Smart Checkout Nexus performs several background tasks that enhance system performance, ensure data integrity, and support a seamless customer experience without disrupting the user interface.

1. **Session Management:**
   The system automatically initiates a session when a customer begins checkout and monitors for inactivity. If no activity is detected after a set period (e.g., 90 seconds), the session times out, clearing all cart data and returning the interface to the Home Screen to protect customer privacy and reset the station for the next user.
2. **Real-Time Cart Synchronization:**
   As users scan items or update the cart (e.g., apply coupons or remove items), the web application syncs data in the background to the backend server and the cloud database. This allows for real-time updates on both the customer interface and the attendant dashboard.
3. **Assistance Call Monitoring:**
   When a customer calls for assistance, the system triggers a backend listener that alerts the attendant device. Background tasks log the call details (station ID, time, and issue type) to track resolution time for quality assurance purposes.
4. **Security Logging:**
   All key actions (e.g., scanning restricted items, payment attempts, overrides) are logged on the server side with timestamps and user session identifiers. These logs support auditing, fraud detection, and compliance requirements.
5. **Scale Weight Polling:**
   The system continuously polls the connected scale in intervals during produce transactions to detect and retrieve accurate weight data, even while other UI elements remain responsive.
6. **Payment Status Verification:**
   For credit/debit payments, a background listener monitors communication from the Verifone terminal. Once the transaction is approved or declined, the result is pushed back to the main application to determine the next UI step without requiring manual refresh or intervention.
7. **System Health Checks:**
   The server periodically runs system health checks on all connected stations and services (e.g., printer, scale, database). Any detected failures are logged and displayed on the attendant dashboard for prompt resolution.

These background operations ensure that the Smart Checkout Nexus operates efficiently and securely, supporting both the customer-facing and back-office needs of a modern retail environment.

Database Model / Schema

Smart Checkout Nexus uses a dual-database integration model to support real-time performance, scalability, and enterprise interoperability. The system interacts with two distinct database platforms:

1. **SAP Database (Enterprise System):**
    - Used for inventory updates, pricing, and financial reconciliation.
    - Integration occurs through secure APIs and scheduled background tasks.
    - Data such as product inventory levels, SKU prices, and financial summaries are synced periodically or on-demand from the web application.
2. **MySQL Database (Operational System):**
    - Handles real-time transactional data and customer-related records.
    - Stores the core operational data needed for running the self-checkout experience.

Key MySQL Tables

- **Customers**
    - customer_id (PK)
    - email
    - loyalty_number
    - opted_in_promotions (boolean)
- **Transactions**
    - transaction_id (PK)
    - customer_id (FK)
    - total_amount
    - payment_method
    - timestamp
    - receipt_type (email, print)
- **Transaction_Items**
    - item_id (PK)
    - transaction_id (FK)
    - product_id
    - quantity
    - weight (nullable)
- **Promotions**
    - promo_id (PK)
    - description
    - discount_type
    - value
    - start_date
    - end_date
- **Assistance_Calls**
    - call_id (PK)
    - station_id
    - reason
    - attendant_id (FK)
    - timestamp

- status

These tables support real-time read/write operations, including cart management, loyalty handling, and interaction logging.

The architecture enables seamless data flow between the customer-facing UI and backend systems, with SAP providing enterprise-level inventory and financial tracking, while MySQL manages real-time operational data for the Smart Checkout Nexus experience.

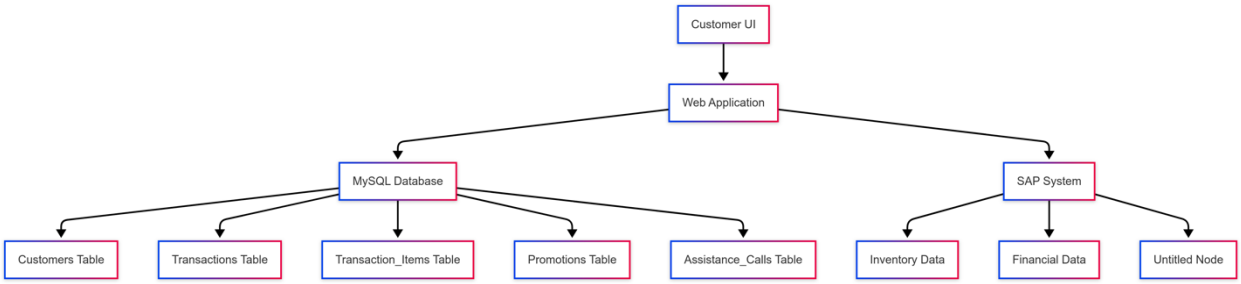

Figure 4 Smart Checkout Nexus – Database Layer Architecture

Client-Side and Server-Side Responsibilities

Smart Checkout Nexus follows a clean separation of concerns between the client and server, promoting scalability, maintainability, and performance optimization. The client-side interface is responsible for rendering user dialogs and handling input, while the server-side logic manages data processing, database communication, and system integrations.

Client-Side Responsibilities

- **UI Rendering:** Dynamically loads and updates user interfaces based on the current transaction state (e.g., scanning, payment, assistance call).
- **Input Handling:** Captures customer interactions such as barcode scans, weight entries, coupon applications, and loyalty number input.
- **State Management:** Maintains session-based cart data and current screen state, which is synchronized with the backend on each interaction.
- **Assistance Triggers:** Sends real-time calls for help or override requests to the attendant dashboard via API requests.
- **Validation & Feedback:** Provides immediate feedback to users for successful scans, restricted items, and invalid entries.

Server-Side Responsibilities

- **API Endpoints:** Handles incoming requests from the client (e.g., `POST /scan`, `POST /apply-coupon`, `GET /product-info`) and responds with processed results.
- **Business Logic:** Determines item pricing, eligibility for discounts, and validation of restricted item overrides.
- **Database Operations:**
    - Writes transactional data to the **MySQL** database in real-time.
    - Syncs inventory and pricing data with **SAP** for financial and stock updates.
- **Payment Processing Integration:** Communicates with the Verifone device to confirm transaction success/failure and updates the server-side transaction record accordingly.
- **Assistance Workflow Logging:** Captures and timestamps assistance requests, attendant responses, and resolutions for reporting.
- **Security & Session Monitoring:** Logs user sessions and critical actions for audit tracking and timeout handling.

This architecture allows the front end to remain fast and responsive while offloading data-heavy tasks to the backend. It also supports scalability by isolating concerns between the UI, business logic, and data layers.

User-Related Tasks and Features

Smart Checkout Nexus implements each user-facing task using a modular design approach. These features are based on the functional requirements identified in the earlier phases of the project and are supported by user interface wireframes created in Milestone 2.

1. Scan Item

Customers begin by scanning an item using the barcode scanner. The system sends a request to the backend, which returns the product information from the SAP system and displays it in the cart view. If a restricted item is scanned (e.g., alcohol), the interface transitions to an override screen until an attendant approves the transaction.
See Figure 5: Home Screen Wireframe
See Figure 6: Cart View Wireframe

2. Produce Lookup and Weighing

If the scanned item is a produce item without a barcode, the customer can search or browse for the item using the produce lookup screen. After selection, the item is weighed automatically using an integrated scale, and the price is calculated based on the weight.
See Figure 7: Produce Lookup Screen Wireframe
See Figure 8: Scale Interaction Screen Wireframe

3. Apply Coupon

Customers can scan or manually enter coupon codes during checkout. These codes are validated against the Promotions table in the MySQL database. If valid, the system recalculates totals and updates the cart.
See Figure 5: Home Screen Wireframe (Coupon Button)
See Figure 6: Cart View Wireframe

4. Enter Loyalty Information

Customers may enter their loyalty number or phone number to retrieve stored data and apply eligible promotions. This information is verified against the Customers table.
See Figure 5: Home Screen Wireframe (Loyalty Info Button)

5. View and Manage Cart

The cart allows customers to review scanned items, update quantities, and remove or rescan items before completing checkout. All changes are tracked in real time.
See Figure 6: Cart View Wireframe

6. Call for Assistance

At any time, customers can request help by selecting the “Call for Assistance” option. This triggers a background task that alerts the attendant’s device.
See Figure 5: Home Screen Wireframe (Assistance Button)
See Figure 9: Attendant Response Wireframe

7. Attendant Functions

When a customer calls for assistance or requires an override, attendants receive a real-time notification on their handheld dashboard. They can acknowledge the call, perform overrides, and manage promotions or look up items.
See Figure 10: Attendant Login Wireframe
See Figure 11: Attendant Dashboard Wireframe
See Figure 12: Override Request Wireframe

8. Payment and Receipt

Customers select a payment method—cash or card—after finalizing their cart. The system interfaces with the Verifone terminal for card transactions and updates financial records accordingly. After payment, users choose to print or email their receipt.
See Figure 13: Receipt Option Wireframe

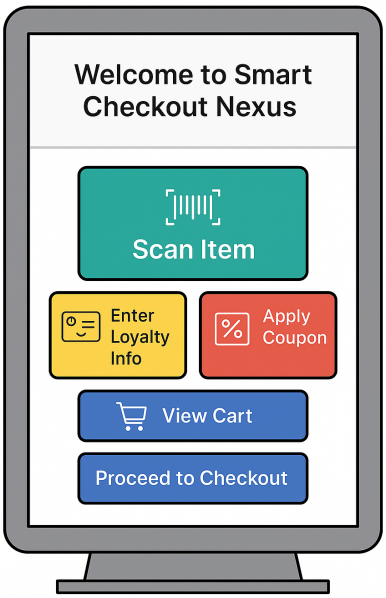

Figure 5 Home Screen Wireframe

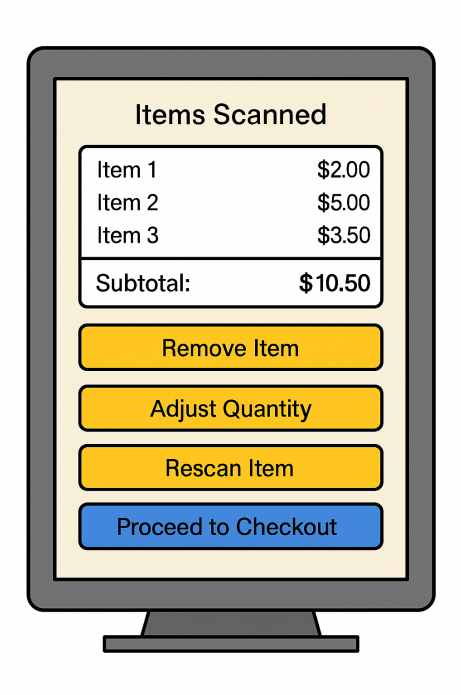

Figure 6 Cart View Wireframe

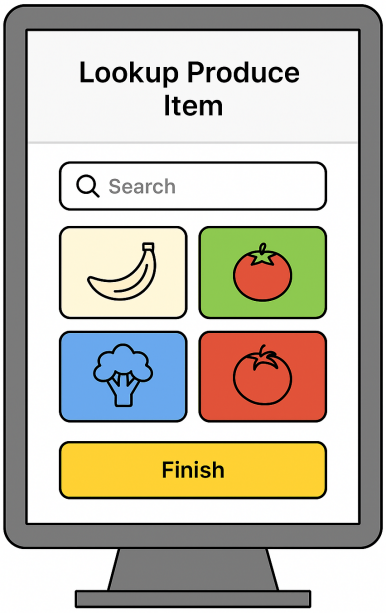

Figure 7 Produce Lookup Wireframe

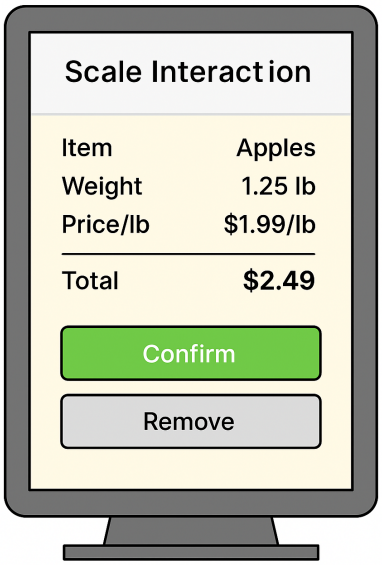

Figure 8 Scale Interaction Wireframe

Figure 9 Attendant Response Wireframe

Figure 10 Attendant Login Wireframe

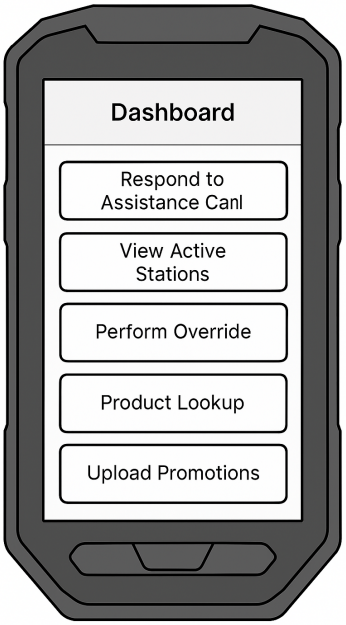

Figure 11 Attendant Dashboard Wireframe

Figure 12 Override Request Wireframe

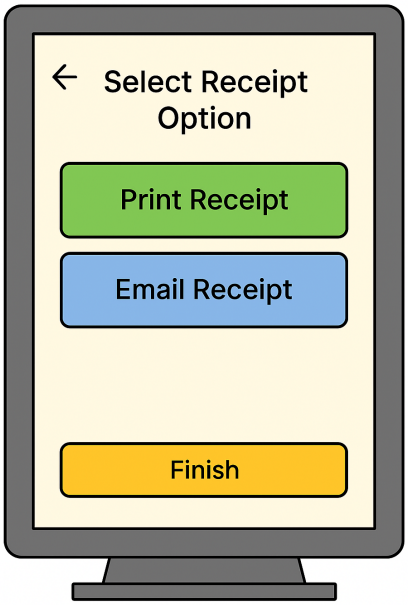

Figure 13 Receipt Option Wireframe

Interfaces with Other Systems

Smart Checkout Nexus integrates with several external systems and hardware devices to ensure seamless retail operations. These interfaces are critical for completing transactions, maintaining inventory accuracy, and complying with financial processing requirements.

1. SAP System (Inventory and Financial Updates)

The web application communicates with the SAP system to:

- Retrieve real-time inventory levels and product pricing
- Post completed transaction summaries for financial reconciliation
- Sync updates on restricted item classifications or category changes

This integration is handled through secure RESTful API calls and scheduled data sync processes between the application server and the SAP system.

2. MySQL Database

While not an external system per se, the MySQL database is treated as a separate component due to its centralized role in managing:

- Customer profiles
- Promotions
- Transaction records
- Assistance call logs

The application uses a secure connection with query-layer abstraction to maintain and fetch data.

3. Verifone Payment Terminal

For credit and debit card transactions, the system interfaces with a Verifone terminal. Once a user selects the payment method, the Verifone device handles:

- Card insertion, tap, or swipe
- PIN entry or signature (if required)
- Communication with the bank or processor for approval

Upon successful authorization, the terminal sends a confirmation to the web app to complete the transaction and trigger the receipt process.

4. Integrated Scale

The scale used during produce transactions connects directly to the web application via a USB or serial interface. The system polls the scale during active weigh interactions, retrieves the weight, and calculates item cost in real time.

5. Receipt Printer

After payment, the system sends a print command to the connected receipt printer for physical receipts. This interface is local and does not require server-side processing unless a reprint or override is triggered from the attendant dashboard.

Each of these integrations is designed to operate asynchronously where possible to preserve UI responsiveness while maintaining real-time functionality.

Statistics Collection for Non-Functional Requirements

To ensure Smart Checkout Nexus meets its non-functional requirements, the system implements several tracking and monitoring mechanisms. These tools and background processes collect data used to evaluate system performance, usability, and compliance over time.

1. Security Logging and Audit Trails

- All critical actions (e.g., login attempts, override approvals, failed payments) are logged with timestamps and user/session IDs.
- Logs are stored in a secure, access-controlled database table.
- These logs can be reviewed by system admins to detect misuse or unauthorized access.

2. Usability Tracking

- The system tracks average transaction times, abandonment rates (e.g., user walks away after scanning), and time spent on individual screens (e.g., scale, payment).
- This data helps identify bottlenecks or confusing UI elements and informs future iterations.
- Feedback prompts or optional usability surveys can be added post-transaction to collect customer satisfaction ratings.

3. ADA and Accessibility Metrics

- Screen reader compatibility is confirmed through accessibility testing tools and manual reviews.
- Button sizes, color contrast, and text labels are designed to meet WCAG 2.1 Level AA standards.
- Usage of accessibility features (e.g., screen magnification or audio prompts) is optionally tracked in anonymous session logs to ensure adequate performance.

4. System Uptime and Performance

- Background health checks log system uptime, error rates, and response times from all active stations.
- Any system crashes, lag spikes, or API failures are timestamped and stored for later diagnostics.
- Performance logs can be used to guide system upgrades or adjust server load balancing strategies.

5. Call for Assistance Metrics

- Assistance calls are timestamped at both request and resolution points.
- This data is used to evaluate average response times and attendant availability.
- Unresolved calls that time out are flagged for supervisor follow-up.

By collecting this data, Smart Checkout Nexus ensures that it not only meets baseline non-functional requirements but also continues to improve its reliability, accessibility, and customer experience over time.

References

Grand Canyon University. (n.d.). *CST-326: Written and Verbal Communication for Software Development - Course Resources*.

MySQL. (n.d.). *MySQL database service*. Oracle. https://www.mysql.com

SAP. (n.d.). *ERP for small and midsize businesses*. SAP. https://www.sap.com

Verifone. (n.d.). *Payment terminals and solutions*. Verifone. https://www.verifone.com

World Wide Web Consortium (W3C). (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. https://www.w3.org/TR/WCAG21/