**Software Requirement Specification (SRS) for Smart Vending Machine**

**Project Name: Smart Vending Machine**

**Date: 06/03/2025**

**Version: 1.0**

**By: Aliyan Ahmed Cheema**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| **Version** | **Author** | **Version Description** | **Date Completed** |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Review history** | | | |
| **Approving party** | **Version approved** | **Signature** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Approval history** | | | |
| **Reviewer** | **Version reviewed** | **Signature** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. Introduction
   * 1.1 Product Scope
   * 1.2 Product Value
   * 1.3 Intended Audience
   * 1.4 Intended Use
   * 1.5 General Description
2. Functional Requirements
3. External Interface Requirements
   * 3.1 User Interface Requirements
   * 3.2 Hardware Interface Requirements
   * 3.3 Software Interface Requirements
   * 3.4 Communication Interface Requirements
4. Non-functional Requirements
   * 4.1 Security
   * 4.2 Capacity
   * 4.3 Compatibility
   * 4.4 Reliability
   * 4.5 Scalability
   * 4.6 Maintainability
   * 4.7 Usability
   * 4.8 Other Non-functional Requirements
5. Definitions and Acronyms

**1. Introduction**

**1.1 Product Scope**

The Smart Vending Machine system will replace traditional vending machines with an intelligent, connected solution. The system will provide a touchscreen-based customer interface, a mobile-based restocking interface, and a desktop-based management interface. This solution aims to reduce human errors in restocking, improve inventory management, provide real-time analytics, and enable centralized control over vending machine operations.

**1.2 Product Value**

This product will:

* Improve operational efficiency by reducing manual data entry and restocking errors.
* Enable real-time inventory and sales tracking.
* Enhance customer experience through an intuitive touchscreen interface.
* Reduce expired and recalled product issues.
* Provide valuable sales analytics for informed marketing decisions.

**1.3 Intended Audience**

* Vending machine operators and companies.
* Restockers responsible for replenishing vending machines.
* Corporate marketing specialists analyzing sales data.
* Software developers and IT administrators managing the system.

**1.4 Intended Use**

* Customers will interact with the touchscreen to make purchases.
* Restockers will use a mobile/tablet interface to receive stocking instructions and scan items.
* Corporate users will access a management tool on a desktop to analyze sales data and send stocking instructions.
* The system will send real-time data to corporate headquarters for efficient decision-making.

**1.5 General Description**

The Smart Vending Machine will consist of:

* A touchscreen interface for customers.
* A mobile-based interface for restockers.
* A management dashboard for corporate use.
* Real-time data transmission for inventory and sales tracking.

**2. Functional Requirements**

* Customers can select and purchase items using a touchscreen.
* The system will provide real-time inventory updates.
* Restockers will receive detailed restocking instructions via a mobile interface.
* The system will track expiration dates and recalled products to prevent errors.
* Corporate users can monitor sales data, track machine status, and push instructions to vending machines.
* The system will support basic analytics for sales trends.
* Machines will report operational status (e.g., offline, low stock, last restocked time).

**3. External Interface Requirements**

**3.1 User Interface Requirements**

* The customer interface must have large, scalable touchscreen buttons.
* The restocker interface should be mobile-friendly.
* The management interface should be desktop-optimized with detailed data visualization.

**3.2 Hardware Interface Requirements**

* The vending machine must support a touchscreen interface.
* Barcode or RFID scanning capabilities for restockers.
* Connectivity (Wi-Fi or cellular) for real-time data transmission.

**3.3 Software Interface Requirements**

* Integration with the company's backend for real-time data exchange.
* Support for future payment methods (e.g., student ID cards, digital wallets).
* Modular design for easy adaptation to different vending machine models.

**3.4 Communication Interface Requirements**

* Real-time data sync between vending machines and corporate servers.
* Notifications for restockers and corporate users on low inventory or machine issues.

**4. Non-functional Requirements**

**4.1 Security**

* Secure authentication for restockers and corporate users.
* Encrypted data transmission.
* Protection against unauthorized access to vending machine data.

**4.2 Capacity**

* The system should handle thousands of vending machines nationwide.
* Each machine should store a local cache of recent transactions in case of connectivity loss.

**4.3 Compatibility**

* Must work with existing vending machine hardware.
* Cross-platform support for management tools (Windows, Mac, and Linux).

**4.4 Reliability**

* The system should continue vending operations even if the management tool is offline.
* Minimal downtime during updates.

**4.5 Scalability**

* Must support additional vending machines as the company expands.
* Should allow future upgrades like AI-driven marketing insights.

**4.6 Maintainability**

* Regular software updates should be easy to deploy.
* Modular architecture to allow easy maintenance and feature expansion.

**4.7 Usability**

* User-friendly interface for all stakeholders.
* Minimal training required for restockers.

**4.8 Other Non-functional Requirements**

* Support for multiple languages if required.
* Touchscreen UI should be adjustable based on screen size.

**5. Definitions and Acronyms**

* **SVM**: Smart Vending Machine
* **UI**: User Interface
* **UX**: User Experience
* **RFID**: Radio Frequency Identification
* **API**: Application Programming Interface