High-Level Design Document for Vending Machine

Contents

[Introduction 1](#_Toc185417650)

[Purpose 1](#_Toc185417651)

[Scope 1](#_Toc185417652)

[Functional Requirements 1](#_Toc185417653)

[Item Selection 1](#_Toc185417654)

[Payment 1](#_Toc185417655)

[Transaction Cancellation 1](#_Toc185417656)

[Unaccepted form of payments 2](#_Toc185417657)

[Item Dispensing 2](#_Toc185417658)

[Change Dispensing 2](#_Toc185417659)

[Inventory Management 2](#_Toc185417660)

[Non-Functional Requirements 2](#_Toc185417661)

[Performance 2](#_Toc185417662)

[Reliability 2](#_Toc185417663)

[System Overview 3](#_Toc185417664)

[System Architecture 3](#_Toc185417665)

[Key Components 3](#_Toc185417666)

[User Interface 3](#_Toc185417667)

[Money Manager 3](#_Toc185417668)

[Inventory Manager 3](#_Toc185417669)

[State Manager 3](#_Toc185417670)

[Error Handling 3](#_Toc185417671)

[Testing Strategy 3](#_Toc185417672)

[Conclusion 3](#_Toc185417673)

# Introduction

## Purpose

This document provides a high-level design for a vending machine system that allows users to select an item, pay for the item, and receive their selected item. The system also handles transaction cancellation, dispenses change, and notifies the system admin about low inventory.

## Scope

The scope of this document includes the design of the vending machine's software components and interactions.

# Functional Requirements

## Item Selection

* The user selects an item by interacting with the user interface.
* The user interface shows the price of each item.
* The user can only select one item per transaction.
* Out of stock items are not shown to the user and cannot be selected.

## Payment

* The machine accepts cash payments only using one-dollar bills, although the system does allow for configuring acceptable denominations.
* Coins are not accepted.

## Transaction Cancellation

* The user may cancel the transaction after making partial payment in which case the user is refunded in the amount of the partial payment.
* The refund is given in coins.
* After the transaction is cancelled, the user may select a different item.

## Unaccepted form of payments

* If the bill is not accepted because it is an unsupported denomination, the bill is returned to the customer.

## Item Dispensing

* The vending machine dispenses the item upon successful payment, i.e. no remaining balance.
* At this point, the user cannot cancel the transaction and get a refund.

## Change Dispensing

* The vending machine calculates and dispenses change if needed.
* Change is dispensed only in coins.

## Inventory Management

* The vending machine tracks item stock levels.
* The user interface shows which items are low or out of stock so that a system admin can take action.
* The default low stock notification threshold is two items left in stock, which is configurable.

# Non-Functional Requirements

## Performance

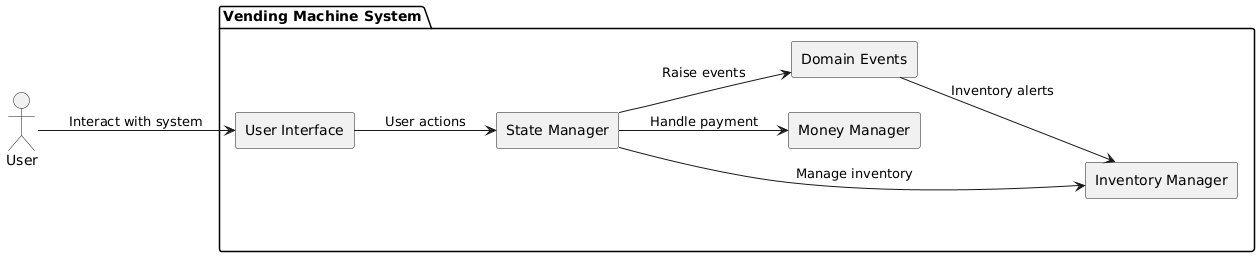
* There are no particular performance requirements.
* The machine will be used by one user at a time. There will be a few dozen users per day.

## Reliability

* There are no particular reliability requirements.
* Inventory will be managed using an in-memory database.

# System Overview

## System Architecture



## Key Components

### User Interface

Allows users to view prices, items, and select an item.

### Money Manager

Handles cash payments, refunds, and change.

### Inventory Manager

Maintains an in-memory database of items, prices, and quantities.

### State Manager

* Coordinates actions between the user interface, money manager, and inventory manager.
* Raises domain events such as low item quantity or out of stock item.

# Error Handling

Invalid actions (e.g., selecting an invalid item, inserting invalid denominations) trigger descriptive error events which bubble to the user interface.

# Testing Strategy

* Integration tests for state transitions and event handling.
* Edge case tests for invalid inputs and out of sequence operations.

# Conclusion

The vending machine system design leverages modular architecture, event-driven communication, and state management to ensure maintainability and extensibility. It provides a solid foundation for building a user-friendly vending experience.