Software Requirements Specification

for

Perishables Management System

Version 1.0 approved

Prepared by Project Managers
Team Members:
Gariman Gangwani - PES1UG21CS202
Joel Renjith - PES1UG21CS247
Himnish N - PES1UG21CS232
Diya Maria Abraham - PES1UG21CS188

PES University, *RR Campus*

15 September 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document outlines the software functional and nonfunctional requirements for the initial release (1.0) of the Perishables Management System (PMS). It is intended for use by the project team responsible for the system's implementation and validation. Unless explicitly stated otherwise, all requirements mentioned here are considered high-priority and are confirmed for inclusion in release 1.0.

1.2 Intended Audience

The intended audience for this Software Requirements Specification of a Perishables Management System (PMS) includes a diverse group of stakeholders involved in the system's development and operation. This audience encompasses the project team, consisting of software developers, business analysts, testers, and project managers, as well as business stakeholders like product owners, executives, and operations managers. Regulatory authorities and external partners such as suppliers, distributors, and retailers also need to understand the system's requirements. Additionally, IT support, maintenance teams, documentation specialists, investors, legal and compliance professionals, auditors, and end consumers may be part of the SRS's audience. The SRS acts as a crucial communication tool, ensuring a shared understanding of the PMS's purpose, features, and constraints among these stakeholders, facilitating effective project planning and implementation.

1.3 Product Scope

The PMS facilitates the management of perishable goods, including tracking, monitoring, and control of stock levels, batch numbers, and expiration dates. The implementation of sensors allows monitoring and control of environmental conditions such as temperature and humidity during storage. PMS involves forecasting demand for perishable products, optimizing inventory levels, and minimizing waste as well as dynamically allocating a discount to facilitate the products getting sold.

1.4 References

To be filled

2. Overall Description

2.1 Product Perspective

The Perishables Management System (PMS) described in this Software Requirements Specification (SRS) is introduced as a standalone product developed to meet the specific challenges associated with managing perishable goods within the organization. While the organization may have existing inventory and supply chain management systems, these systems often lack the specialized features necessary for effective perishable product management. The PMS offers a dedicated solution for the end-to-end monitoring and control of perishables, from suppliers to consumers. To enhance understanding, a diagram depicting the main components of the broader system, their interactions, and external interfaces is included in the SRS to illustrate how the PMS integrates into the organization's operational framework.

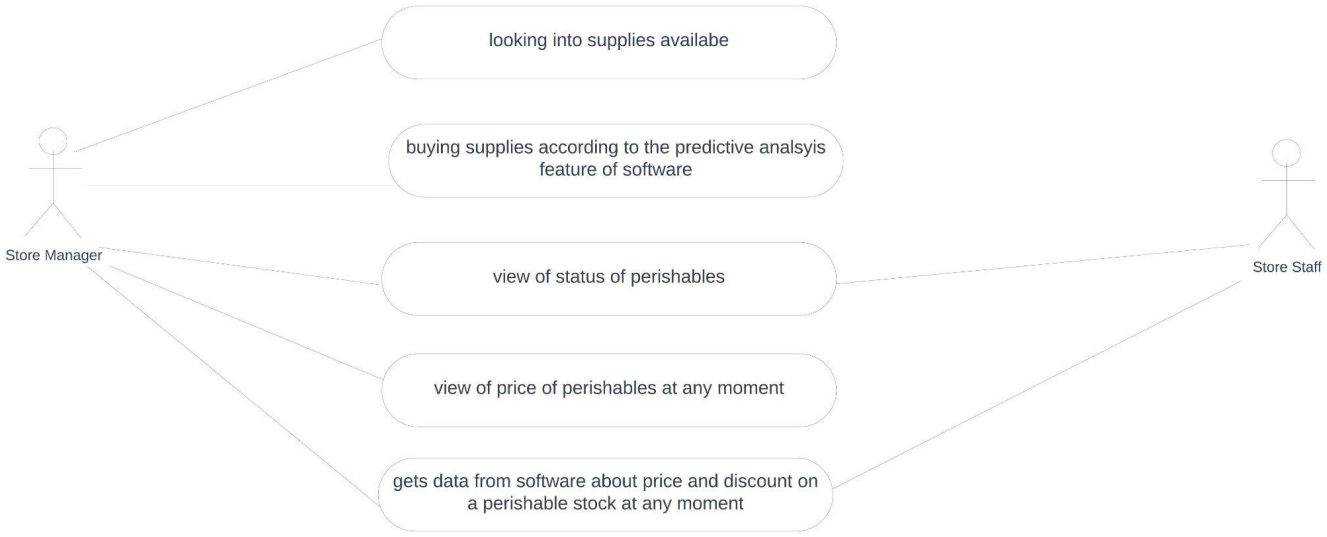
2.2 Product Functions

The Perishables Management System (PMS) is designed to perform a range of essential functions for the efficient management of perishable goods. These functions include inventory management to track stock levels and monitor product expiration, order processing to handle orders and prioritize them based on product freshness,temperature and environmental monitoring, demand forecasting, and robust reporting and analytics capabilities. It also suggests discounts that can be placed on the perishable items to facilitate it being bought and reducing the waste generated. These functions collectively enable the system to support the entire lifecycle of perishable products and contribute to minimizing waste.

2.3 User Classes and Characteristics

- **User Class: Store Manager**
  Characteristics:
  - *View Stock Availability*: The store manager can access real-time information regarding the available stock from buyers within the system.
  - *Purchase Supplies*: The store manager has the capability to make informed purchasing decisions based on recommendations generated by the predictive analysis of perishable item expiration dates within the software.
  - *Monitor Perishable Status*: The system provides the store manager with vital details regarding perishable items, including their remaining shelf life, the environment they are stored in, and the current temperature conditions.
  - *Check Perishable Prices*: At any given moment, the store manager can instantly retrieve the current prices of perishable items stored in the system.
  - *Access Discount Information*: The software furnishes the store manager with up-to-date information on any discounts or promotional offers applicable to perishable stock within the system.
- **User Class: Store Staff**
  Characteristics:
  - *Monitor Perishable Status*: Store staff members have the ability to monitor the status of perishable items, including information about their remaining shelf life, the environment they are stored in, and the current temperature conditions.
  - *Prioritize Stock Sales*: The system provides store staff with data indicating which perishable items must be sold earliest. This information enables them to arrange and prioritize stock placement within the store accordingly to minimize waste and optimize sales.
  - *Access Discount Information*: Store staff can access real-time data from the software regarding any discounts or promotions available for perishable items. This helps them

provide customers with accurate pricing information and maximize sales opportunities.



2.4 Operating Environment

- OE-1: Web Browser Interface:
    - The software will primarily operate through a web browser interface, providing users, especially the store manager, with access to intuitive dashboards and a comprehensive view of inventory.
        - The backend will be developed using Django, while the front end will utilize React JS, and the database of choice will be MySQL.
    - This web-based platform facilitates efficient monitoring and management of perishables, making it a valuable tool for the store manager user class.
- OE-2: Mobile Application:
    - In addition to the web browser interface, the software will also be available as a mobile application.
        - The Android version will be developed using Kotlin, while the iOS app will be created using Swift.
    - This mobile app ensures that data is conveniently accessible from anywhere, catering to the needs of the store staff user class.
    - Store staff can easily check perishable status, prioritize stock sales, and access discount information on the go, enhancing their ability to manage perishable inventory effectively.

2.5 Design and Implementation Constraints

To be filled

2.6 User Documentation

- The system shall provide an online hierarchical and cross-linked help system in HTML that describes and illustrates all system functions.
- User documentation starts with an introduction, including an overview of the system and its intended audience.
- The documentation covers topics such as system setup, user roles, basic and advanced operations, troubleshooting, best practices and security.

2.7 Assumptions and Dependencies

AS-1 : It is assumed that users will have a stable internet connection for real-time data access and updates. In the absence of a reliable internet connection, system performance and data synchronization may be affected.

DE-1 : The operation of the discount-assignment component of PMS depends on changes being made in the Perishables Inventory System to update the availability of food items as they get sold.

DE-2 : External environmental conditions, such as power outages or natural disasters, can impact the operation of hardware components used for monitoring. Contingency plans for such events should be considered.

3. External Interface Requirements

3.1 User Interfaces

UI-1:The interface uses graphical icons, buttons, menus, and other elements of a graphical user interface (GUI) to allow users to interact with the system.

UI-2:For example, the interface could use tabs, breadcrumbs, or sidebars to organize the content into categories and subcategories.

UI-3:For example, the interface could use labels, tooltips, progress bars, or notifications to display feedback, instructions, warnings, or errors.

3.2 Hardware Interfaces

HI-1: RFID tags and readers.

HI-1.1: RFID (Radio Frequency Identification) tags are small devices that can be attached to perishable items and store information such as product name, expiry date, temperature, and location.

HI-1.2: The technology can help monitor the inventory levels, track the movement of items, and alert the users about any spoilage or theft.

HI-2: Sensors and actuators can help maintain the optimal conditions for perishable items, detect any anomalies or faults, and automate the warehouse operations.

HI-2.1: The sensors can measure physical parameters such as temperature, humidity, pressure, light, and sound.

HI-2.2: The actuators can perform physical actions such as opening or closing a valve, turning on or off a fan, or adjusting a thermostat.

HI-3: Touchscreens and keyboards can be used to provide input controls and informational components for the users of the perishable inventory management system.

HI-3.1: Touchscreens can display graphical user interfaces (GUIs) and allow users to interact with them by touching the screen.

HI-3.2: Keyboards are devices that allow users to enter text or commands by pressing keys.

3.3 Software Interfaces

SI-1: The GUI should have a clear and intuitive navigation system that helps users find the information and functions they need

SI-1.1: A graphical user interface (GUI) that allows users to view, enter, edit, and delete information about their perishable inventory, such as product name, quantity, expiry date, location, etc.

SI-1.2: The GUI should provideS features such as search, filter, sort, print, export, import, etc. to help users manage their inventory efficiently and effectively.

SI-1.2: The GUI should have a clear and intuitive navigation system that helps users find the information and functions they need.

SI-1.3: GUI could use tabs, breadcrumbs, or sidebars to organize the content into categories and subcategories.

SI-2: A database interface that allows the software system to store and retrieve data from a database server.

SI-2.1: The database interface should ensure the security, integrity, consistency, and availability of the data.

SI-2.2: The database interface should also support transactions, concurrency control, backup and recovery mechanisms.

3.4 Communications Interfaces

External Interfaces:

Web Service Interface: Uses HTTP and JSON for real-time data exchange with external systems, following RESTful principles.
Database Interface: Interacts with a specified DBMS using SQL queries and a defined data model.

Internal Interfaces:

Graphical User Interface (GUI): Utilizes HTML, CSS, and JavaScript to provide an intuitive, user-friendly interface with various input controls and data presentation elements.
Hardware Interface: Communicates with physical devices and sensors using specific communication protocols and data formats, along with standard drivers or libraries.

Assumptions and Dependencies:

Assumption: Users have access to a stable internet connection.

Dependency: The operation of the discount-assignment component relies on real-time updates from the Perishables Inventory System.

Dependency: The hardware interface is dependent on the availability and proper functioning of physical devices and sensors.

Contingency Plans:

Communication Failure: In the event of a communication failure due to internet connectivity issues:

- Contingency Plan: Implement a local caching mechanism to temporarily store critical data and sync with the central system once the connection is restored.

Dependency Disruption (Discount Assignment): If the PMS experiences downtime or data synchronization issues:

- Contingency Plan: Implement a manual override option for discount assignment, allowing authorized personnel to make adjustments temporarily.

Hardware Component Failure: In case of hardware component failures or sensor malfunctions:

- Contingency Plan: Regularly monitor the status of hardware components and establish a maintenance schedule to quickly replace or repair faulty components.

Power Outages or Natural Disasters: External environmental conditions such as power outages or natural disasters impacting hardware components:

- Contingency Plan: Implement backup power sources (e.g., uninterruptible power supplies) and establish disaster recovery procedures to ensure minimal system downtime.

4. System Features

4.1

4.1.1 Description and Priority

This feature leverages historical purchase data and item expiry rates to determine optimal restocking points based on item expiration dates and the current stock levels, taking recent purchase trends into consideration. Priority = High

4.1.2 Stimulus/Response Sequences

Stimulus: The store manager accesses their dashboard and clicks on the notification button.

Response: The system provides stock purchase recommendations through predictive analysis via notifications.

Stimulus: The store manager receives an email from the software containing stock purchase recommendations generated through predictive analysis. The email includes a button for further information and can also serve as a reminder to restock.

Response: The store manager clicks the button, leading them to a website that provides a comprehensive and detailed view of the recommendations.

4.1.3 Functional Requirements

To be filled

4.2 Discount Allocation

4.2.1 Description and Priority

This feature involves the allocation of discounts and promotions for perishable products within the PMS. It allows for the implementation of various discount strategies, such as volume discounts and time-based promotions.The priority for this feature is considered medium because while discount allocation is important for boosting sales and managing perishable inventory effectively, it may not be as critical as core functionalities like inventory management or quality control.

4.2.2 Stimulus/Response Sequences

Stimulus - Perishables present in large quantities in the inventory, not being sold.
Response - Allocate discount to perishables. Assign larger discounts to perishables that are present in higher quantity and closer to expiring. Periodically increment this discount percentage as expiry date approaches.

4.2.3 Functional Requirements

To be filled

4.3 Perishables Catalog

4.3.1 Description and Priority:

This interface serves as a catalog for perishable items, allowing users to access a comprehensive list of all perishables available in the shop. Users can conveniently view essential information such as cost, discount, lifetime (time until expiry), as well as temperature and humidity details for each perishable item.

4.3.2 Stimulus/Response Sequences

Stimulus: User clicks the "Perishable Catalog" button in the navigation bar.
Response: The system displays a comprehensive catalog featuring all the perishable items available in the shop. Each item within the catalog is accompanied by an image, its price, the current discount, and a dropdown menu for further actions.

Stimulus: User clicks on the dropdown menu associated with an item in the catalog.
Response: The dropdown expands to reveal a row-wise presentation of additional information, including the shelf life, available stock quantity, temperature, and humidity details of the selected perishable item.

4.3.3 Functional Requirements

To be filled

5. Non-functional Requirements

5.1 Performance Requirements

The PMS must be available 99% of the time. Any downtime can potentially result in loss of revenue for the store. The system shall display confirmation messages to users within 4 seconds after the user submits information to the system.

5.2 Safety Requirements

Not Applicable

5.3 Security Requirements

The system should have robust security measures in place to protect sensitive data related to perishable items and their transactions.

5.4 Software Quality Attributes

Not Applicable

5.5 Business Rules

Not Applicable

Appendix A: Glossary

To be filled

Appendix B: Analysis Models

To be filled

Appendix C: To Be Determined List

To be fill