

Software Requirement Specification for Mess Groceries Stock Maintenance

Name	Deepak S
Roll no	7376222IT124
Seat no	297
Project ID	17
Problem Statement	Mess Groceries Stock Maintenance

1. Introduction

In the realm of institutional or organizational setups such as mess facilities, effective management of groceries is pivotal to ensure smooth operations and optimal resource utilization. This document outlines the specifications for a system designed to facilitate the maintenance of groceries in a mess environment.

1.1. Purpose:

The purpose of this document is to delineate the design and functionality of a software system tailored specifically for managing groceries within a mess facility. It aims to streamline the process of procurement, storage, issuance, consumption tracking, and reporting, thereby enhancing efficiency and reducing operational overheads.

1.2. Scope of Project:

The project scope includes developing a groceries stock management system with features like indent management, order processing, receipt tracking, sub-store management, consumption recording, expiry tracking, reporting, user management, and a user-friendly interface. It aims to streamline stock operations, enhance visibility, and ensure efficient management of groceries stock.

2. System Overview:

- **Indent Raise Entry:**

Users should be able to log in and raise indents for the groceries they need.

Include fields for item name, quantity, unit, date needed, etc.

- **Approval for the Indent:**

Approvers should receive notifications for pending indents.

They can review the indents and either approve or reject them.

- **Raise the Order against the Indent:**

Once an indent is approved, it should be converted into an order automatically or by the user.

Include fields for vendor selection, delivery date, etc.

- **Receipt of Groceries with Bill Details:**

Users should be able to log the receipt of groceries along with the bill details.

Upload feature for attaching scanned copies of bills.

- **Sub-Store Wise Issuance of Groceries:**

Ability to issue groceries from the main stock to sub-stores.

Tracking issuance details like date, quantity, sub-store, etc.

- **Consumption of Groceries on Each Mess and Session Wise:**

Record consumption data for each mess (kitchen) and session (breakfast, lunch, dinner).

Include fields for date, item, quantity used, etc.

- **Expiry Product Details:**

Maintain a database of products with expiry dates.

Automated alerts for nearing expiry products.

- **Reports for Stock Maintenance:**

Generate reports on current stock levels.

Alerts for low-stock items.

Historical data analysis for stock trends.

- **Monthly Reports:**

Monthly summaries of stock movements, consumption, orders placed, etc.

- **Cost Report:**

Calculate the cost of groceries consumed over a specific period.

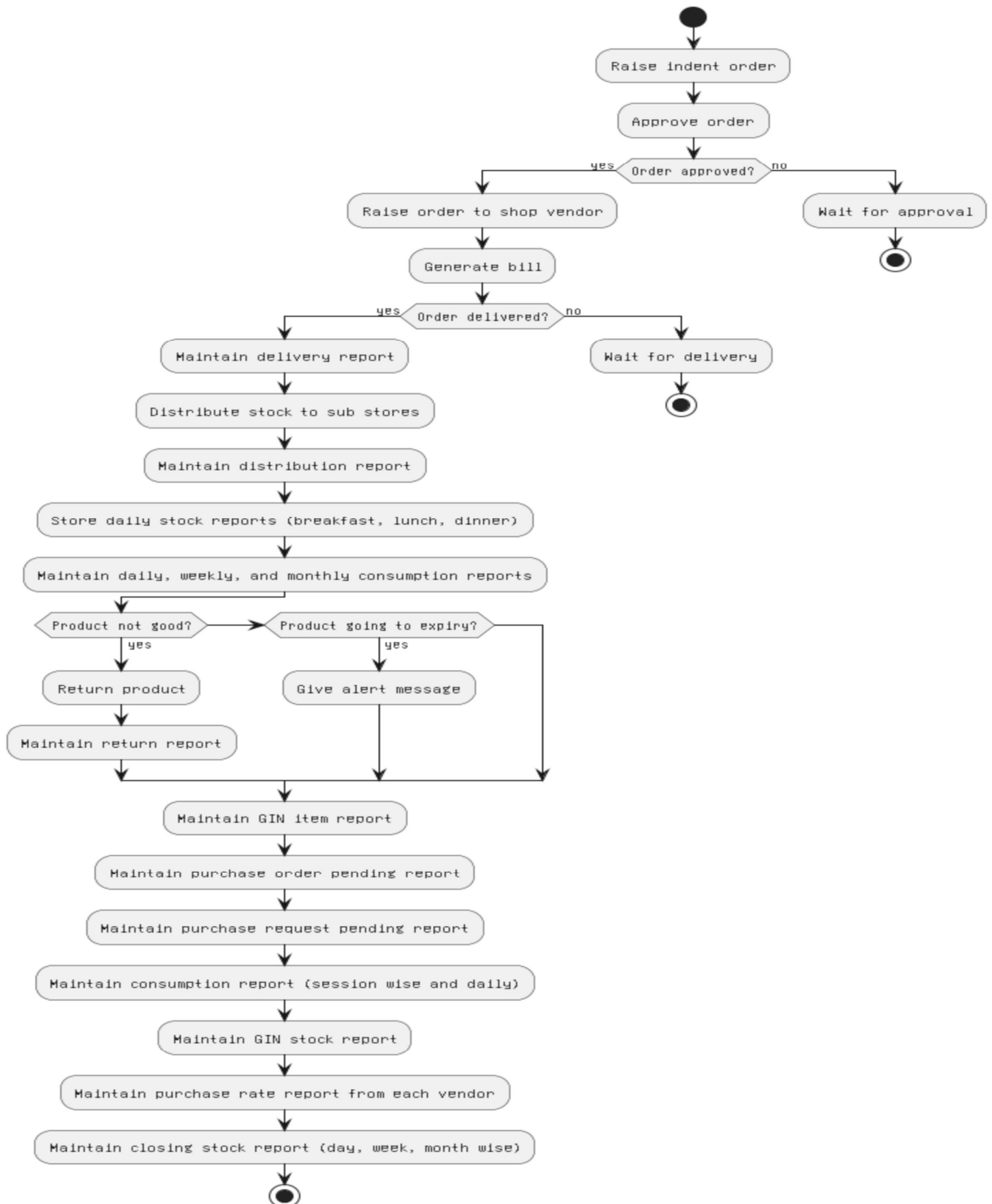
Compare costs with budgeted amounts.

- **Comparison Rate Charts of Groceries:**

Visualize data using charts to compare prices over time or between different vendors.

Show trends in consumption and costs.

Work flow:



3.1. Functional Requirements;

User Management:

User roles include mess managers, staff, and administrators.

Authentication and authorization mechanisms to control access to functionalities.

Inventory Management:

Add, edit, and delete grocery items with details such as name, quantity, unit, and expiry date.

Track stock levels and receive alerts for low-stock items or nearing expiry products.

Procurement:

Place orders with vendors for replenishing stock.

Vendor management functionalities for maintaining vendor details and communication.

Issuance:

Issue groceries from the main stock to sub-stores or directly to end-users.

Track issuance details including date, quantity, and recipient.

Consumption Tracking:

Record consumption data for each mess and session, capturing details such as date, item, and quantity used.

Reporting and Analysis:

Generate reports on stock maintenance, consumption, orders placed, and cost analysis.

Provide visualizations such as charts to facilitate data interpretation.

3.2. Non-Functional Requirements:

Performance:

Ensure system responsiveness within acceptable time frames, even under peak loads.

Optimize database queries and server-side processing for efficient performance.

Security:

Implement robust authentication and authorization mechanisms to safeguard sensitive data.

Encrypt data during transmission and storage to prevent unauthorized access.

Usability:

Design an intuitive user interface with clear navigation and informative error messages.

Provide user training and documentation to facilitate ease of use.

Reliability:

Ensure high system availability with minimal downtime for uninterrupted operations.

Implement backup and recovery mechanisms to mitigate data loss risks.

Scalability:


Design the system to accommodate growing data volumes and user loads over time.


Ensure modular architecture to support future enhancements and scalability.

Admin Interface

Admin Dashboard


Manage Users

Add User 


View/Edit User 


Delete User 

Manage Orders

View Orders 

Approve/Reject Orders 

Logout 

Generate Reports 

Manage Vendors


Add Vendor 


View/Edit Vendor 


Delete Vendor 


User Interface

User Dashboard

View Profile 


Place Order 

Track Order 

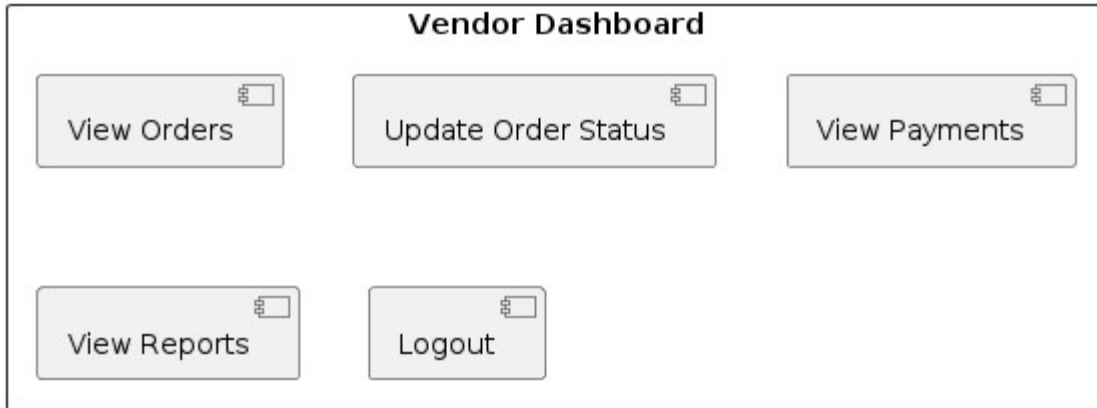
View Consumption 

View Notifications 

Settings 

Logout 

Vendor Interface



Stack:

Front End	Html, Css, Javascript
Backend	Python, Django (Web)
Data Base	Mysql, Postgresql