**Software Requirement Specification for Mess Groceries Stock Maintenance**

| **Name** | Surya Aswaththan G |
| --- | --- |
| **Roll no** | 7376222AL213 |
| **Seat no** | 177 |
| **Project ID** | 17 |
| **Problem Statement** | Infra-Mess groceries stock maintenance |

**1**. **Introduction**

**1.1. Purpose:**

The purpose of this document is to present a detailed description of the Mess groceries stock maintenance portal. It will explain the purpose and features of the portal interface of the system. This system aims to streamline the process of purchase request submission, approval, procurement, and stock management, while providing comprehensive dashboards for different user roles.

**1.2. Scope of Project:**

● This software system will serve as a portal for the Mess groceries stock maintenance, enabling staff to submit Purchase requests of stocks for their mess. From an administrative perspective, this system will provide a comprehensive **analytical dashboard for project oversight**.

● Administrator (Manager) has the ability to approve or reject Purchase requests (PR). Once a Purchase Request is approved, staff can view the status of the Purchase Order using their dashboard. The system will calculate the stock reports based on the data provided on consumptions of stocks. If the stock level is below 30%, a warning will be displayed indicating that stocks are used to minimum point and to refill stock , which is contingent on the number of stocks left.

**2. System Overview:**

**2.1. Users:**

1. **Manager**:

They have the ability to approve or reject a purchase request, monitor the status of the stock level, consumption of stocks in statistical values, and review their transaction and purchase history.

**2. Admins:**

Review submitted purchase requests, manage stocks based on category and create (or) add a stock to the portal, and access analytical dashboards for stock oversight.

**2.2. Features:**

**1. Login and registration:**

Manager, Admins, Staff can register for an account or login with their existing account

**2. Purchase Request Submission:**

Staff can input relevant details regarding their purchase of stock including stock name, quantity, mess detail, date of request, and any necessary attachments. Upon completion, the purchase request is submitted to the manager/admin interface for review and further processing

**3. Application Status:**

Manager can view the current status of the stock level, consumption level, transaction, return stock report, pending reports and requests, and many, and also see the history logs in the option Activity

**4. Purchase Order:**

Once the manager approves the Purchase requests, admins have the ability to place order on the requested stocks to respected vendors with necessary details.

**5. Manager Access:**

Manager(s) can view all submitted Purchase requests applications of stock category wise, view application details, approve or reject the application with suitable remarks, and view all the statistical data on stock usage, consumption,etc…

**5. Admin Access:**

Admin can view all submitted purchase request applications in a category of stock, view application details, once purchase request is approved by manager, they have access to proceed further on purchase order of a stock based on purchase requests from all mess.

**6. Manager’s Analytical Dashboard:**

Manager(s) can view the number of applications by category, purchase requests and can also see the log of pending approvals , history of logs(reports), statistical reports on stock consumption, purchase history, vendor wise cost report, day-wise cost report, purchase report, bill reports, closing stock reports and more.

**3. System Requirements Specification:**

**3.1 Functional Requirements:**

**● User Management:**

○ Staff can login to the portal using their credentials.

○ Admins have access control with an analytical dashboard and dedicated features like create (or) add (or) delete (category based) .

○ Manager(s) have access control with an analytical dashboard and dedicated features like approve or reject purchase requests.

**● Purchase Report / Purchase Order Application:**

○ Staffs can submit purchase request applications with appropriate details

○ Purchase Request Application form contains:

■ Stock name

■ Category of the stock

■ Quantity

■ Mess detail

■ Date of request

○ Purchase Order Application form contains:

■ Stock name

■ Category of the stock

■ Quantity

■ Vendor detail

■ Price ($ cost)

**● Application Status:**

○ Staff can view the current status of their purchase request application

○ If the application is rejected then the remarks is shown

○ Staff can also see the logs of their applications

○ Managers can view all reports, data, and approve/reject PRs. Admins manage item and supplier master data and handle inward purchases.

**● Admin Dashboard:**

○ Admins can view a list of all submitted Purchase request applications.

○ Applications can be filtered by category (mess,type of stock). ○ Admins can view details of each application.

○ Admins can proceed on purchase order with suitable approval.

○ Admins can create,add, or delete stock and more details.

**● Analytics Dashboard:**

○ Admin can view the number of applications by its category

○ Manager(s) can view Purchase requests (also pending) which are requested based on the category.

**3.2. Non-Functional Requirements:**

● **Performance**: The system must respond to user actions within 2 seconds to ensure efficient usability and the system should perform efficiently, even with a large volume of data.

● **Security**: Robust security measures should be implemented to protect user data and prevent unauthorized access.User data must be encrypted during transmission and storage, and access to sensitive functionalities should be restricted to authorized admin users through secure authentication mechanisms.

● **Usability**: The interface should be intuitive and user-friendly for all user roles, with clear and concise error messages provided to guide users in case of input errors or system failures.

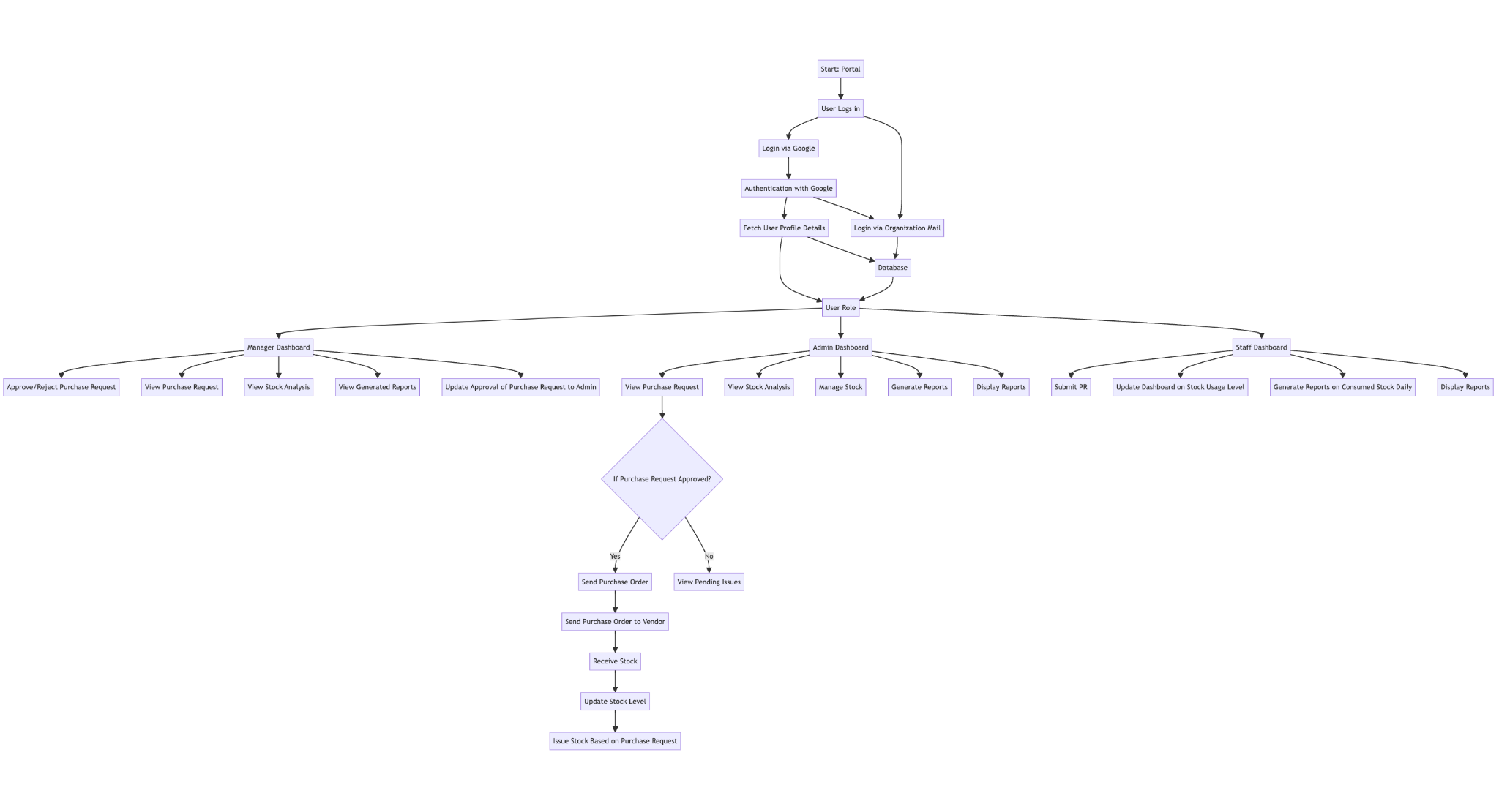
● **Reliability**: The system should be reliable and available 24/7 with minimal downtime and should have a backup and recovery mechanism in place to prevent data loss in case of system failures or crashes.

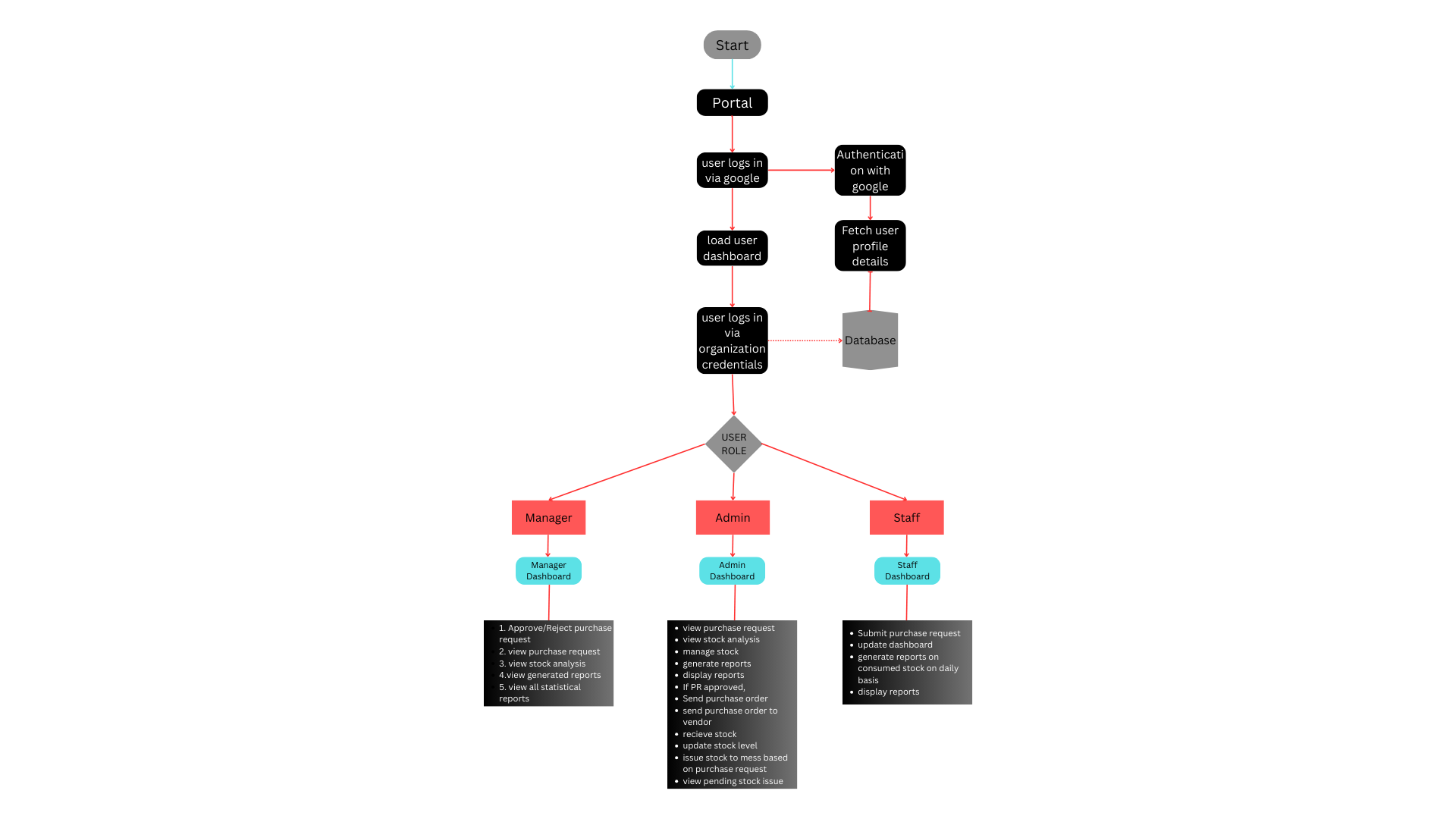
● **Scalability**: The system should be designed to accommodate an increasing number of users and data volume over time, and the system should be scalable to accommodate future growth and changes in user requirements.

**Stack:**

| Front End | HTML, CSS, JavaScript |
| --- | --- |
| Backend | Apache web server, PHP with Laravel Framework |
| Data Base | MySQL |

**Flow Chart**

****

****