MediSupply Use Cases

1. Overview

MediSupply is a database-driven Django web application designed to streamline the purchase order process for healthcare institutions. The system enables hospital staff to create purchase orders for medical supplies, allows a purchasing manager to review and approve these orders, and permits suppliers to update order status (e.g., shipped, delivered). Additional administrative functions support user and system management. This document details all use cases that our project will implement.

2. Actors

- **Hospital Staff:**
  Initiates new purchase orders and views their order history.
- **Purchasing Manager:**
  Reviews purchase orders submitted by Hospital Staff, approves or rejects orders, and can request modifications.
- **Supplier:**
  Views approved purchase orders and updates the order status (e.g., shipped or delivered).
- **Administrator:**
  Manages system settings, user roles, and overall configuration (basic administration).

3. Use Cases

3.1 Create Purchase Order

- **Primary Actor:** Hospital Staff
- **Preconditions:**
  - User must be authenticated and have the Hospital Staff role.
  - The user has access to the "Create Purchase Order" page.
- **Basic Flow:**
  1. Hospital Staff logs into the system.
  2. Navigates to the "Create Purchase Order" section.
  3. Enters purchase order details (order date, list of items, quantities, prices, etc.).
  4. Submits the form.
- **Alternate Flow:**
  - If required fields are missing, the system displays an error message and prevents submission.
- **Postconditions:**
  - A new purchase order is created with an initial status of "Pending" and stored in the database.

3.2 Approve/Reject Purchase Order

- **Primary Actor:** Purchasing Manager
- **Preconditions:**
  - A purchase order exists with a "Pending" status.
  - The Purchasing Manager is logged in.

- **Basic Flow:**
  1. The manager accesses the dashboard displaying pending orders.
  2. Selects a purchase order to review.
  3. Reviews the order details.
  4. Chooses to either **Approve** or **Reject** the order.
- **Alternate Flow:**
  - If the order details are incomplete or the request does not meet necessary criteria, the Manager may request modifications from the Hospital Staff before making a final decision.
- **Postconditions:**
  - The purchase order’s status is updated to "Approved" or "Rejected".
  - Notification is sent to the originating Hospital Staff user.

3.3 Update Order Status

- **Primary Actor:** Supplier
- **Preconditions:**
  - The purchase order is in an approved state.
  - The Supplier is authenticated in the system.
- **Basic Flow:**
  1. The supplier logs in and navigates to the list of approved orders.
  2. Selects an order to update.
  3. Updates the order status to "Shipped" (or directly "Delivered", if applicable).
- **Alternate Flow:**
  - If the update fails (e.g., due to technical issues), an error is logged and a retry message is displayed.
- **Postconditions:**
  - The order status is updated accordingly and the change is reflected in the order history.
  - Notification is sent to both Hospital Staff and Purchasing Manager.

3.4 View Purchase Orders

- **Primary Actors:** Hospital Staff, Purchasing Manager, and Supplier
- **Preconditions:**
  - The user is logged in and has the appropriate role.
- **Basic Flow:**
  1. The user selects the "View Orders" page.
  2. The system retrieves and displays a list of purchase orders based on the user's role:
     - **Hospital Staff:** Sees orders they have created.
     - **Purchasing Manager:** Sees pending orders for review.
     - **Supplier:** Sees orders approved for fulfillment.
- **Alternate Flow:**
  - If no orders match the criteria, the system displays a "No orders found" message.
- **Postconditions:**
  - The user can view current orders, including their statuses and key details.

3.5 Additional (Optional) Use Case: User Authentication

- **Primary Actor:** Any user (Hospital Staff, Manager, Supplier, Admin)
- **Preconditions:**
  - The user must have a valid account.
- **Basic Flow:**
  1. The user navigates to the login page.
  2. Provides valid credentials.
  3. Gains access to the system.
- **Alternate Flow:**
  - Invalid credentials cause an error message prompting the user to retry.
- **Postconditions:**
  - A successful login grants role-specific access to the system's functionalities.

End of Use Case Documentation