Here’s a detailed explanation of the **system flows** in your Django e-commerce platform with **role-based access control (RBAC)** and dashboards:

**🧭 System Overview**

The system supports four user roles:

* **Admin**
* **Manager**
* **Staff**
* **Customer**

Each role has a distinct dashboard and access permissions, ensuring users interact only with the features relevant to their responsibilities.

**🔄 System Flows by Role**

**👤 Customer Flow**

1. **Visit Welcome Page** → Sees product categories (Meat, Egg, Cassava Processed, Fruits & Juices Transformed, Vegetables).
2. **Select Category** → Views products in that category.
3. **Browse Products** → Can view product details and add items to cart.
4. **View Cart** → Updates quantities, removes items.
5. **Checkout** → Enters personal data, reviews total cost.
6. **Submit Order** → Order is saved, email confirmation sent.
7. **Order Confirmation Page** → Displays unique reference number and order summary.
8. **Dashboard Access** → Views order history and account details.

**🧑‍💼 Manager Flow**

1. **Login** → Redirected to Manager Dashboard.
2. **Dashboard Features**:
   * View inventory levels.
   * Manage product listings and categories.
   * Monitor order statuses.
   * Access analytics (e.g., top-selling products).
3. **Restricted Access** → Cannot manage users or system settings.

**👷 Staff Flow**

1. **Login** → Redirected to Staff Dashboard.
2. **Dashboard Features**:
   * View pending orders.
   * Update order status (e.g., processing, completed).
   * Manage stock levels.
3. **Restricted Access** → Cannot modify products or access analytics.

**🛡️ Admin Flow**

1. **Login** → Redirected to Admin Dashboard.
2. **Dashboard Features**:
   * Full access to Django admin panel.
   * Manage users and assign roles.
   * Create/edit/delete products and categories.
   * View all orders and system logs.
   * Configure site-wide settings.
3. **Superuser Privileges** → Can override any restrictions.

**🔐 RBAC Implementation**

**✅ Decorators**

* role\_required('manager') → Restricts view to managers only.
* roles\_required('admin', 'manager') → Allows access to both roles.

**✅ Dashboard Routing**

@login\_required

def dashboard(request):

role = request.user.role

if role == 'admin':

return render(request, 'catalog/dashboard\_admin.html')

elif role == 'manager':

return render(request, 'catalog/dashboard\_manager.html')

elif role == 'staff':

return render(request, 'catalog/dashboard\_staff.html')

else:

return render(request, 'catalog/dashboard\_customer.html')

**📊 Dashboard UI Summary**

| **Role** | **Dashboard Template** | **Key Features** |
| --- | --- | --- |
| Admin | dashboard\_admin.html | Full control: users, products, orders |
| Manager | dashboard\_manager.html | Inventory, product management, analytics |
| Staff | dashboard\_staff.html | Order fulfillment, stock updates |
| Customer | dashboard\_customer.html | Order history, account settings |

Would you like a visual flowchart of these system flows or help integrating real-time data into each dashboard?