

# Technical Documentation

## 1. Define technical Requirement

### 1. Frontend (Next.js & Tailwind CSS):

- Matches the customer-facing actions, including product browsing, cart handling, and placing orders.
- Admin views for product and order management fit into this architecture.

### 2. Backend (Sanity CMS):

- Manages customer data, product catalogs, and order history.
- Tracks changes based on admin updates and payment statuses.

### 3. APIs:

- Integrates functionalities like fetching product data, processing orders, and providing delivery tracking.

### 4. Payment Gateway:

- Supports both "Online Payment" and "Cash on Delivery" options.

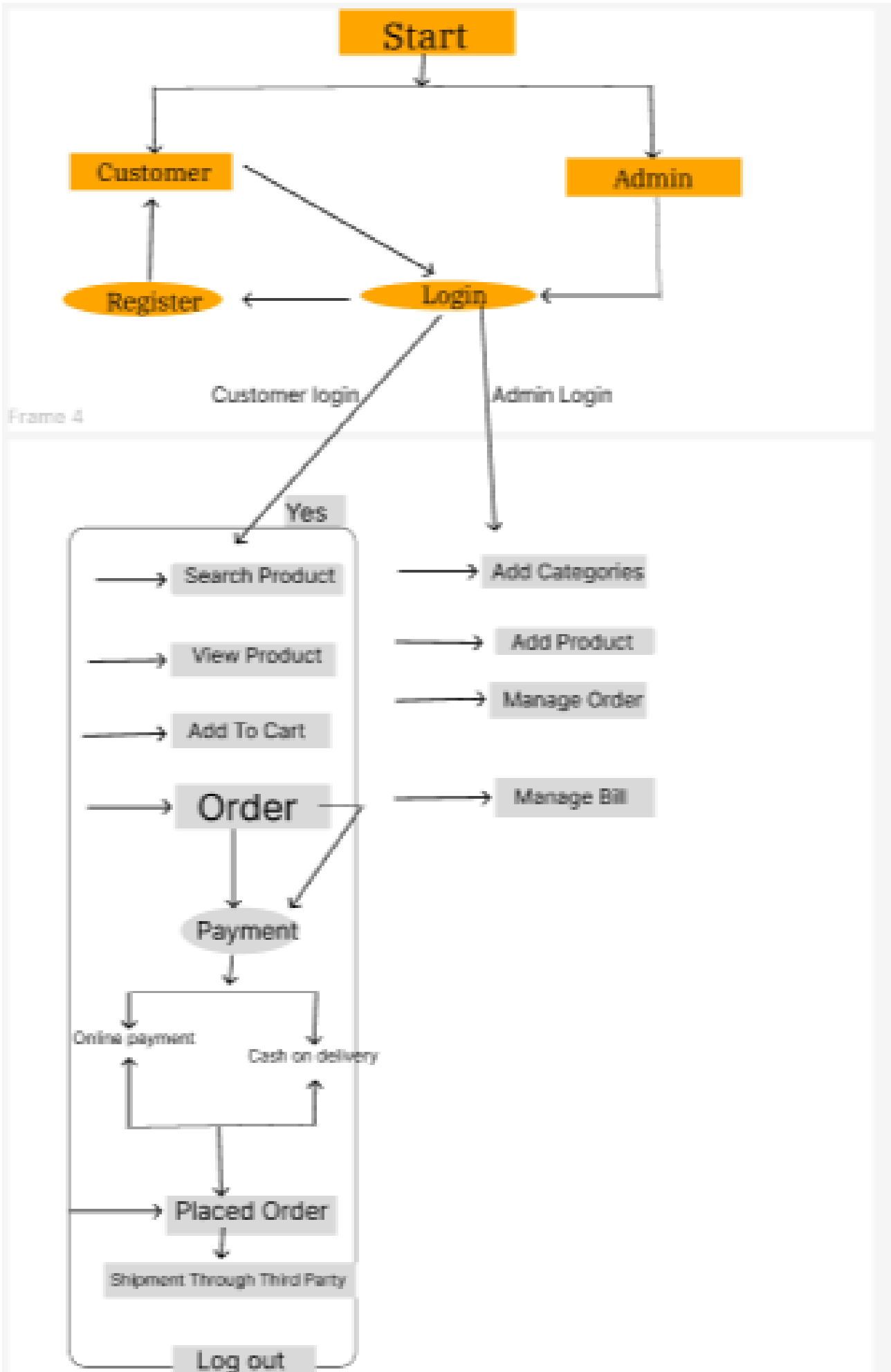
### 5. Delivery API:

- Provides real-time updates for customer orders.

### 6. Technical Roadmap:

- Maps the daily progress for building the described functionalities, ensuring alignment with the described workflows.

# 2.Design System Architecture



# Diagram Description

The system begins with two user types:

1. Customer
2. Admin

Workflow Breakdown:

## 1. Login/Register:

- Customers can register and log in.
- Admins log in directly for system management.

## 2. Customer Functionality: After login, customers can:

- Search Products: Access the product catalog.
- View Products: Explore product details.
- Add to Cart: Select items for purchase.
- Order: Place an order after selecting items.
- Payment Options:
  - Online Payment: Secure online transactions.
  - Cash on Delivery: Pay at the time of delivery.
- Order Status: After payment, the order status updates in real time.
- Log Out: Users can exit their session.

## 3. Admin Functionality: After login, admins can:

- Add Categories: Organize product categories.
- Add Products: Add new products to the system.
- Manage Orders: Oversee orders placed by customers.
- Manage Bills: Handle payment records and financial transactions.