

This document describes the user journey flow an e Commerce market-place,focusing on key steps and their technical implementation.it is tailored for building a robust and user-friendly platform.

Flowchart Overview:

Below is the complete user journey flow for an e Commerce marketplace:

Home Page:

User lands on the homepage.

Display featured categories,popular products,and a search bar.

Product Browsing:

User selects a category or use the search bar.

products are displayed with options to filter by price,brand,or rating.

Product Details:

User clicks on a product to view its details.

Page includes product description,price,availability,and user reviews.

Add to cart:

User adds the product to the cart.

Cart updates dynamically with quantity and price.

Checkout:

Page-3

User proceeds to the checkout page.

Provides shipping address and selects a delivery option.

Payment:

User enters payment details.

Secure payment gateway processes the transaction.

Order Confirmation:

Order details are displayed and sent via email.

Order status is updated in the back-end.

Shipment Tracking:

User visits the "Order History" section.

Real-time shipment tracking is enabled via API integration.

Delivery:

Product is delivered to the user's address.

Front-end Requirements:

User receives a notification and can leave a review.

User-Friendly interface for browsing products.

Responsive design for mobile and desktop users.

Essential pages: Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation.

2. Design System Architecture

Diagram:

[Front-end (Next.js)]



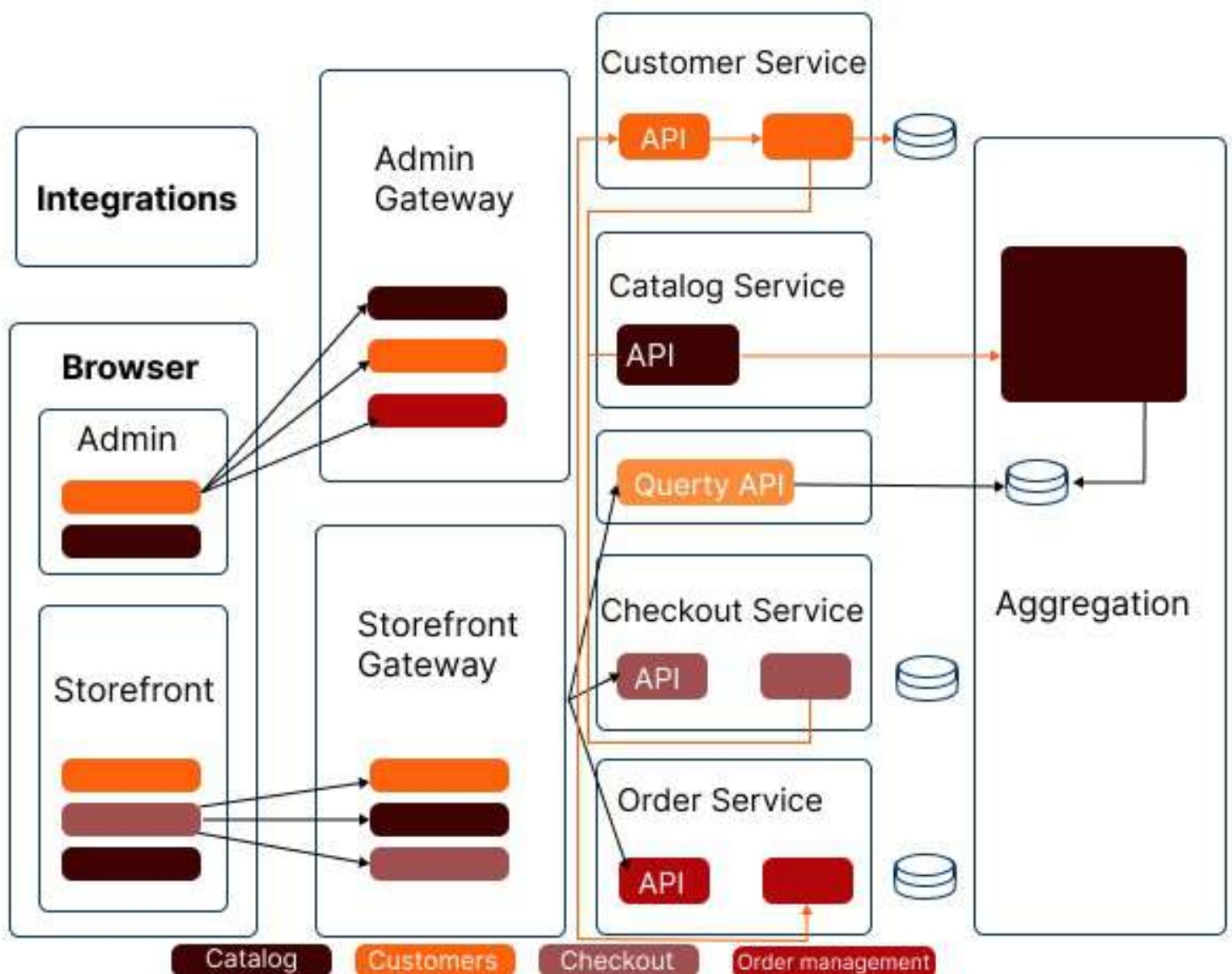
[Sanity CMS] → [Product Data API]



[Third-Party API] → [Shipment Tracking API]



[Payment Gateway]



1. Display the user interface for browsing products, managing the cart, and placing orders.
2. Handles user interaction and communicates with back-end services via APIs.

Sanity CMS:

1. Acts as the primary back-end to manage product data, customer details, and order records.
2. Provides APIs for the front-end to fetch and update data.

Product Data API:

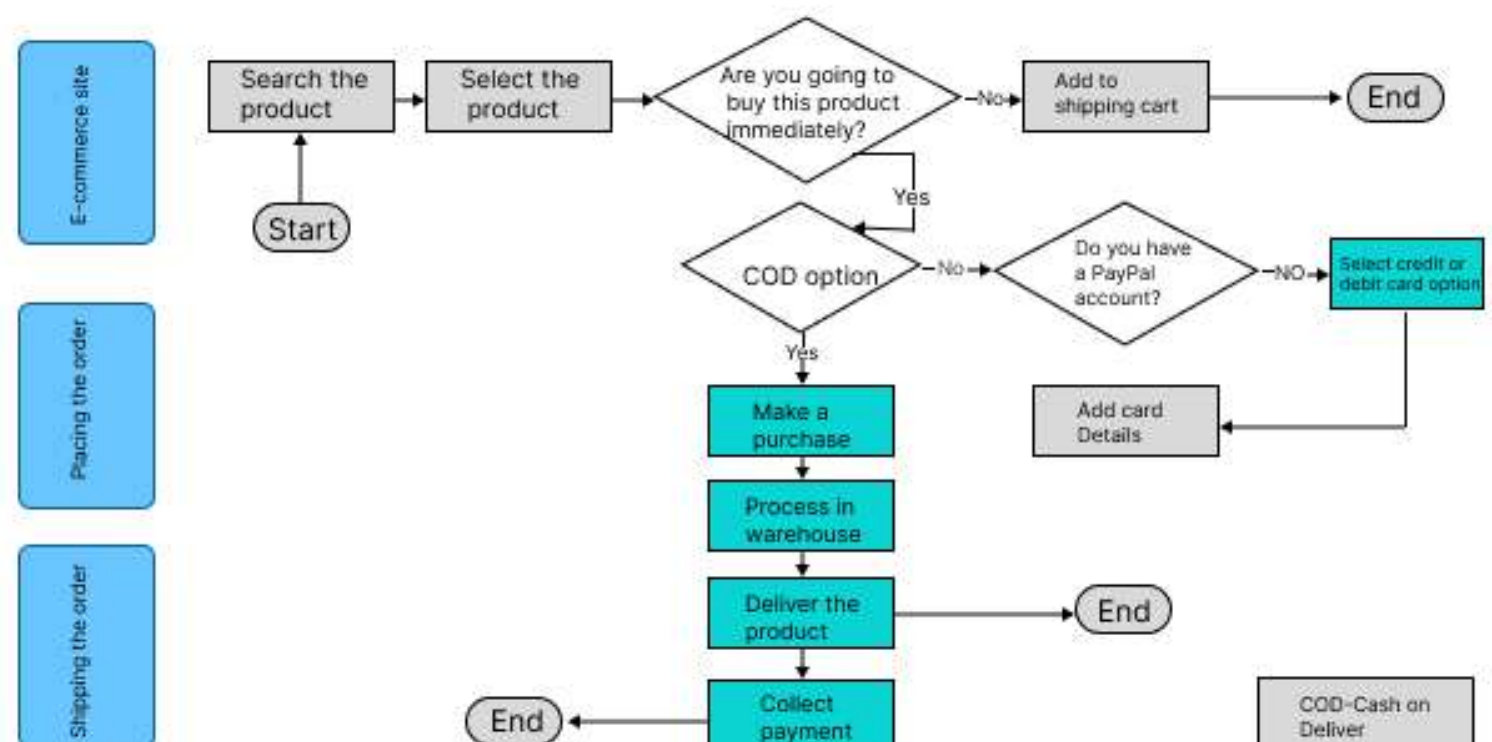
Provides endpoints to fetch product listings, details, and inventory status.

Third-Party APIs:

Integrates services like shipment tracking and payment processing.

Payment Gateway:

Processes user payments securely and provides transaction confirmation.



Below diagram will define the relationships between entities in database.

Example Entities:

1.Furniture Items:

Fields: id, name, price, description,image, categoryId.

2.Categories:

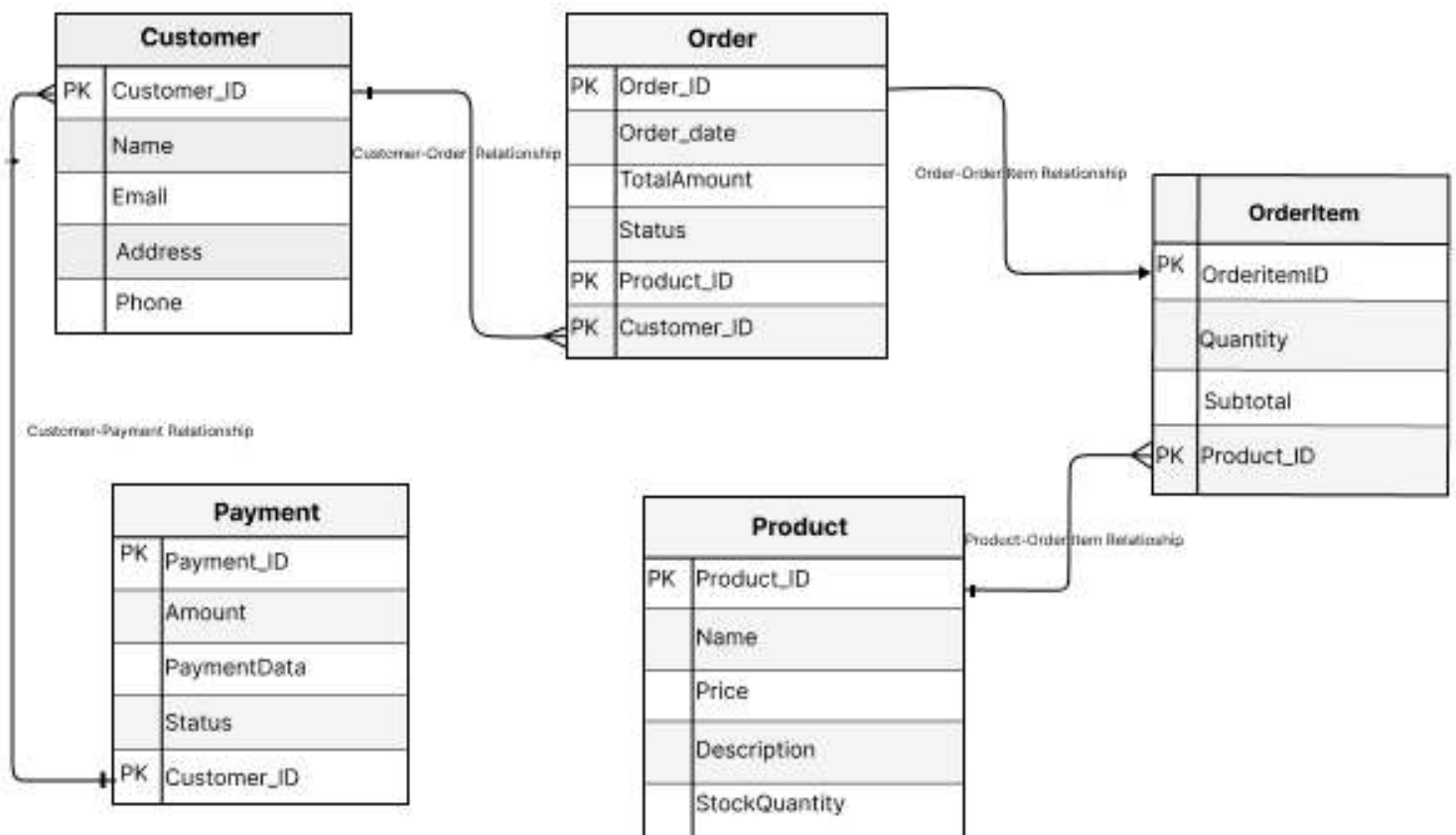
Fields: id, name.

3.Users:

Fields: id, name, email,password.

4.Orders:

Fields: id,userId, orderData, status



Endpoint	Method	Description	Parameters	Response Example
/api/furniture	GET	Fetch all furniture items	None	{id: 1,name:"Table"}
/api/furniture/:id	GET	Fetch a single furniture item	id (Path)	{id: 1,name:"Sofa"}
/api/furniture	POST	Add a new furniture item	name,price,category(Body)	{success:true,id:5}
/api/furniture/:id	PUT	Update a furniture item	id (Path).name.price(Body)	{success:true}
/api/furniture/:id	DELETE	Delete a furniture item	id(Path)	{success:true}
/api/categories	GET	Fetch all furniture categories	None	{categories:["Chair"]}

Workflow Diagram Visual Representation:



1. Product Schema

```

export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'slug', type: 'slug', title: 'Slug', options: { source: 'name', maxLength: 96 } },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'description', type: 'text', title: 'Description' },
    {
      name: 'category',
      type: 'reference',
      title: 'Category',
      to: [{ type: 'category' }]
    },
    { name: 'images', type: 'array', title: 'Product Images', of: [{ type: 'image' }] },
    {
      name: 'vendor',
      type: 'reference',
      title: 'Vendor',
      to: [{ type: 'vendor' }]
    },
    { name: 'isFeatured', type: 'boolean', title: 'Featured Product' }
  ]
};

```

2. Category Schema

```
export default {
  name: 'category',
  type: 'document',
  title: 'Category',
  fields: [
    { name: 'name', type: 'string', title: 'Category Name' },
    { name: 'slug', type: 'slug', title: 'Slug', options: { source: 'name', maxLength: 96 } },
    { name: 'description', type: 'text', title: 'Description' }
  ]
};
```

3. Vendor Schema

```
export default {
  name: 'vendor',
  type: 'document',
  title: 'Vendor',
  fields: [
    { name: 'name', type: 'string', title: 'Vendor Name' },
    { name: 'email', type: 'string', title: 'Vendor Email' },
    { name: 'phone', type: 'string', title: 'Vendor Phone' },
    { name: 'address', type: 'text', title: 'Vendor Address' },
    { name: 'products', type: 'array', title: 'Products', of: [{ type: 'reference', to: [{ type: 'product' }] }] }
  ]
};
```

4. Order Schema

```
export default {
  name: 'order',
  type: 'document',
  title: 'Order',
  fields: [
    { name: 'orderNumber', type: 'string', title: 'Order Number' },
    { name: 'customerName', type: 'string', title: 'Customer Name' },
    { name: 'email', type: 'string', title: 'Customer Email' },
    { name: 'phone', type: 'string', title: 'Customer Phone' },
    { name: 'shippingAddress', type: 'text', title: 'Shipping Address' },
    {
      name: 'products',
      type: 'array',
      title: 'Ordered Products',
      of: [{ type: 'reference', to: [{ type: 'product' }] }]
    },
    { name: 'totalAmount', type: 'number', title: 'Total Amount' },
    { name: 'status', type: 'string', title: 'Order Status', options: { list: ['Pending', 'Processing', 'Shipped', 'Delivered'] } },
    { name: 'orderDate', type: 'datetime', title: 'Order Date' }
  ]
};
```


To track customer profiles:

```
export default {  
  name: 'customer',  
  type: 'document',  
  title: 'Customer',  
  fields: [  
    { name: 'name', type: 'string', title: 'Customer Name' },  
    { name: 'email', type: 'string', title: 'Customer Email' },  
    { name: 'phone', type: 'string', title: 'Customer Phone' },  
    { name: 'address', type: 'text', title: 'Customer Address' },  
    { name: 'orders', type: 'array', title: 'Orders', of: [{ type: 'reference', to: [{ type: 'order' }] }] }  
  ]  
};
```