Domain Driven Design Document

[Shopping Cart System]

I. Introduction

A. Purpose/Objective

(Define the goals and primary objective or mission of this DDD documentation for Shopping Cart System)

B. Domain Scope

(Define the area/end users for the application)

C. Domain Description

(Describing the complete domain briefly including core business domains and functionalities)

D. Context

(Explain the boundaries of this analysis)

II. Strategic Design

A. Bounded Contexts

a. **Define core bounded contexts within our shopping cart system domain**
   Examples:
   Product (product info, product catalog, product attributes/categories)
   Customer/User Management (user profiles, accounts and interactions)
   Sales
   Inventory
   Shipping
   Accounting

b. **Every Contexts should be explained with these as follows:**
   a. Describe its specific responsibility and boundaries
   b. Define Ubiquitous language use within that context
   c. Identify key domain concepts and their specific meanings within the context

B. Context Map

a. **Illustrate all relationships between the above defined bounded contexts\**

b. **Specify the proper type of relationships (customer-supplier, conformist, shared kernel)**

c. **Provide proper brief document for interactions using the relationships.**
   Example: The Sales Context (D) depends on the Product Context for product information in customer-supplier relationships of Conformist.

C. Sub-Domains

a. **Identify the core, supporting and generic subdomains within the shopping cart system**
   Ex: Tabular or Document or visually represented

b. **Give documents for identified domains that helps to prioritize development efforts and determine the proper appropriate strategies followed to be.**

III. **Tactical Design**

A. **Product Context**

a. **Entities**

Product
Category
Attribute
ProductImage

b. **Value Objects**

Price
Quantity
Dimensions

c. **Aggregates/ aggregate root**

Product (Aggregate->Product_ID)

d. **Domain Services**

ProductCategoryService
ProductSearchService

e. **Domain Events**

ProductAdd
ProductPriceChanged
ProductOutOfStock

f. **Repositories**

ProductRepository

g. **Factories**

ProductFactory

h. **Application Services**

ProductManagementService(createProduct(), updateProduct(), getProductdetails())

B. **Customer Service**

a) **Entities**
b) **Value Objects**
c) **Aggregates/ aggregate root**
d) **Domain Services**
e) **Domain Events**
f) **Repositories**
g) **Factories**
h) **Application Services**

C.

D.

IV. **Implementation Considerations**

a. **Technology Considerations**

Technology Discussions (Spring Boot, Angular)
Frameworks (DDD)
Messaging System (RabbitMQ/Kafka)
Specific Data Persistence (ORM/OXM)

b. **Architectural Pattern**
   Event Sourcing (Producer and Consumer)
   SAGA (Transactions)
   CQRS (Which Persistent Store)
   Resilience Patterns: CircuitBreaker Pattern, RateLimiting (Controller Methods/Rest API)
   Deployment: canary / BlueGreen patterns (Microservices)

c. **Testing**
   Unit Testing (Junit5 & Jasmine)
   Integration Testing (Mockito)
   Domain Specific Testing (Stub)
   Security Testing, Slice Testing, Repository Testing

d. **Security**
   AuthN and AuthZ (LDAP, OAuth2.0, VMware Identity Manager/Workspace One)
   Security using JWT, Web Security, Method Security

V. **Glossary**

a. Define all the terminologies, key terms, conceptual definitions in the Ubiquitous Language use across the Bounded Contexts
b. Clarify any potentially or mandate ambiguous terms
c. Illustrative Snippets for Tactical Design Objects

Example Illustrative Snippets:
Product Context – Entity
Entity: Product
Description: ---
Atttributes:

- Product_ID
- Product_Name
- Product_Desc
- Product_Price
- Product_Category
- Product_Image

Sales Context – Aggregate:
Aggregate:
Description:
Entities:
Context Map : Product Context

- **Relationships**
- **Description**
- **Integration: API calls to retrieve product details, product creation, product update**