**Application: Payment Systems Application**  
**I. Introduction**  
**Purpose:** Define the goals and scope of this DDD documentation for the payment systems application  
**Goals:**  
**Scope:**  
**Domain Overview:**  
Briefly describe the payment systems domain, including core business functionalities / tasks  
(ex: transaction processing, account management, payment gateway, invoicing, reconciliation)  
**Context:** Explain the boundaries of the particular model specific to the sub-domains

**II. Strategic Design**  
**A. Bounded Contexts**  
**Define the core bounded contexts within the payment systems application**  
**Account Service:**  
Summary/Functionalities  
**Transaction Processing:**  
Summary/Functionalities  
**Payment Gateway:**  
Summary/Functionalities  
**Invoicing:**  
Summary/Functionalities  
**User Management:**  
Summary/Functionalities  
**For each identified bounded context**  
Describe the specific responsibility and boundaries  
Define the Ubiquitous Language used for the context  
Idenitfy the key domain concepts

**B. Context Map**  
Diagrammatic representation for Context and its relationships  
Specify the relationships (C-S, Conformist, Shared Kernel)  
Document the interactions and data flow between contexts  
  Ex: Transaction Processing Context (D) depends on the Account Context (U) for account information

**C. Sub-Domains**  
Identify the core, supporting and generic subdomains within the payment systems domain  
Prioritize the development efforts and determine the appropriate strategies

**III. Tactical Design**  
**A. Account Service**  
**Entities**  
Account  
AccountHolder  
**Value Objects**  
AccountNumber  
Balance  
Currency  
**Aggregates**  
Account (with holder info, balance details)  
**Domain Services**  
AccountValidationService  
AccountHistoryService  
**Domain Events**  
AccountCreated  
AccountUpdated  
BalanceAdjusted  
**Repositories**  
AccountRepository  
**Factories**  
AccountFactory  
**Application services**  
AccountManagementService( createAccount(), updateAccount(), getAccountDetails())

**B. Transaction Processing**  
**C. Payment Gateway**  
**D. Invoicing**  
**E. User Management**

**IV. Implementation Considerations**  
**Technology choices**  
Discuss technologies and frameworks that align with the DD design (ORM, messaging)  
**Architectural Patterns**  
Describe patterns (CQRS, SAGA, CircuitBreaker, Event Sourcing)  
**Testing Strategy**  
Outline the testing strategies: Unit, integration, slice, repository, web and method testing, security testing, actuator testing

**V. Glossary**  
Define all the terminologies we used in Ubiquitous Language across the bounded contexts  
**Example Code Snippets (illustrative |**