**Application: Healthcare Management Application**

**I. Introduction**

**Purpose**: Define the goals and scope of this DDD documentation for the healthcare management application

**Goals**:

**Scope:**

**Domain Overview**:

Briefly describe the healthcare domain, including core business functionalities / tasks

(ex: patient management, appointment booking, doctor management, treatment, billing)

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 healthcare application

**Patient Service:**

Summary/Functionalities

**Appointment**:

Summary/Functionalities

Medical Records:

Summary/Functionalities

**Billing**:

Summary/Functionalities

Us**er 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: Appointment Context (D) depends on the Patient Context (U) for patient information

**C. Sub-Domains**

Identify the core, supporting and generic subdomains within the healthcare domain

Prioritize the development efforts and determine the appropriate strategies

**III. Tactical Design**

**A. Patient Service**

**Entities**

Patient

EmergencyContact

**Value Objects**

Address

ContactInfo

DateOfBirth

**Aggregates**

Patient (with contact info, emergency contacts)

**Domain Services**

PatientSearchService

PatientHistoryService

Domain Events

PatientRegistered

PatientUpdated

EmergencyContactAdded

**Repositories**

PatientRepository

**Factories**

PatientFactory

**Application services**

PatientManagementService( registerPatient(), updatePatient(), getPatientDetails())

**B. Appointment**

**C. Medical Records**

**D. Billing**

**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 |