

# Technical Implementation Draft for Kind Health System

Saransh Sharma

Version 1.0  
April 4, 2024

## 1 Introduction

This document outlines the technical implementation strategy for the Kind Health system, focusing on leveraging Large Language Models (LLMs) to enhance patient care through a multi-agent, modular architecture. Our approach emphasizes adaptability, modularity, and scalability, ensuring the system remains at the forefront of healthcare technology.

## 2 System Architecture

The system is designed around a core set of principles: modularity, reusability, and loose coupling, enabling flexibility and ease of maintenance. The architecture comprises several key components:

- **Agent Framework:** Utilizes Langroid for creating a robust multi-agent environment.
- **Evaluation Engine:** Incorporates Deep Eval SDK for dynamic, rule-based system evaluation.
- **Tracing Mechanism:** A custom-built tracing system providing transparency and explainability.
- **User Interface:** An intuitive UI integrating system functionalities for end-users, particularly healthcare professionals.

## 3 Agent Framework

The agent framework is the backbone of our system, facilitating the interaction between different autonomous agents, each with specific roles and responsibilities.

### 3.1 Agent Classes

```
class Agent:
    def __init__(self, name, role):
        self.name = name
        self.role = role
        self.tasks = []

    def receive_task(self, task):
        self.tasks.append(task)

    def execute_tasks(self):
        for task in self.tasks:
            # Task execution logic
            pass
```

### 3.2 Agent Types

- **Root Agent:** Oversees other agents, ensuring ethical guidelines are followed.
- **Delegation Agent:** Specializes in medical terminology and communication.
- **Doctor Agent:** Stores and manages medical knowledge and feedback.
- **Retrieval Augmented Generation Agent:** Enhances system capabilities by integrating various data sources.

## 4 Evaluation Engine

The evaluation engine is critical for maintaining system accuracy, relevance, and efficiency.

### 4.1 Evaluation Metrics

```
class EvaluationMetric:
    def __init__(self, name, threshold, calculation_method):
        self.name = name
        self.threshold = threshold
        self.calculation_method = calculation_method

    def evaluate(self, data):
        # Evaluation logic based on calculation_method
        return score
```

## 4.2 Sample Metrics

- Explainability and Trust
- Efficiency in Patient Encounters
- Concordance with Physician Decisions

## 5 Tracing Mechanism

The tracing mechanism provides visibility into the decision-making process, crucial for debugging, optimization, and explainability.

### 5.1 Tracing Implementation

```
class Tracer:
    def __init__(self):
        self.traces = []

    def add_trace(self, trace):
        self.traces.append(trace)

    def get_trace(self, identifier):
        # Retrieve specific trace
        return trace
```

## 6 User Interface

The user interface ties together the system components, offering a seamless experience for healthcare professionals.

### 6.1 UI Components

```
class Dashboard:
    def display_patient_data(self, patient_id):
        # Display patient-specific data and insights

class AgentControlPanel:
    def monitor_agent_status(self):
        # Real-time monitoring of agent activities

class EvaluationReport:
    def generate_report(self, metrics):
        # Generate evaluation reports based on selected metrics
```

## 7 Conclusion

This document provides a foundational blueprint for the technical implementation of the Kind Health system. The modular and scalable architecture ensures the system can evolve with technological advancements and changing healthcare needs.