Technical Design Document: Conversational AI Health Assistant

Executive Summary

This document outlines a production-ready architecture for a GenAI-based conversational health assistant that meets all specified requirements: personalized engagement, natural interaction, actionable insights, and extensible architecture using modern AI agent patterns.

Requirements Compliance Checklist

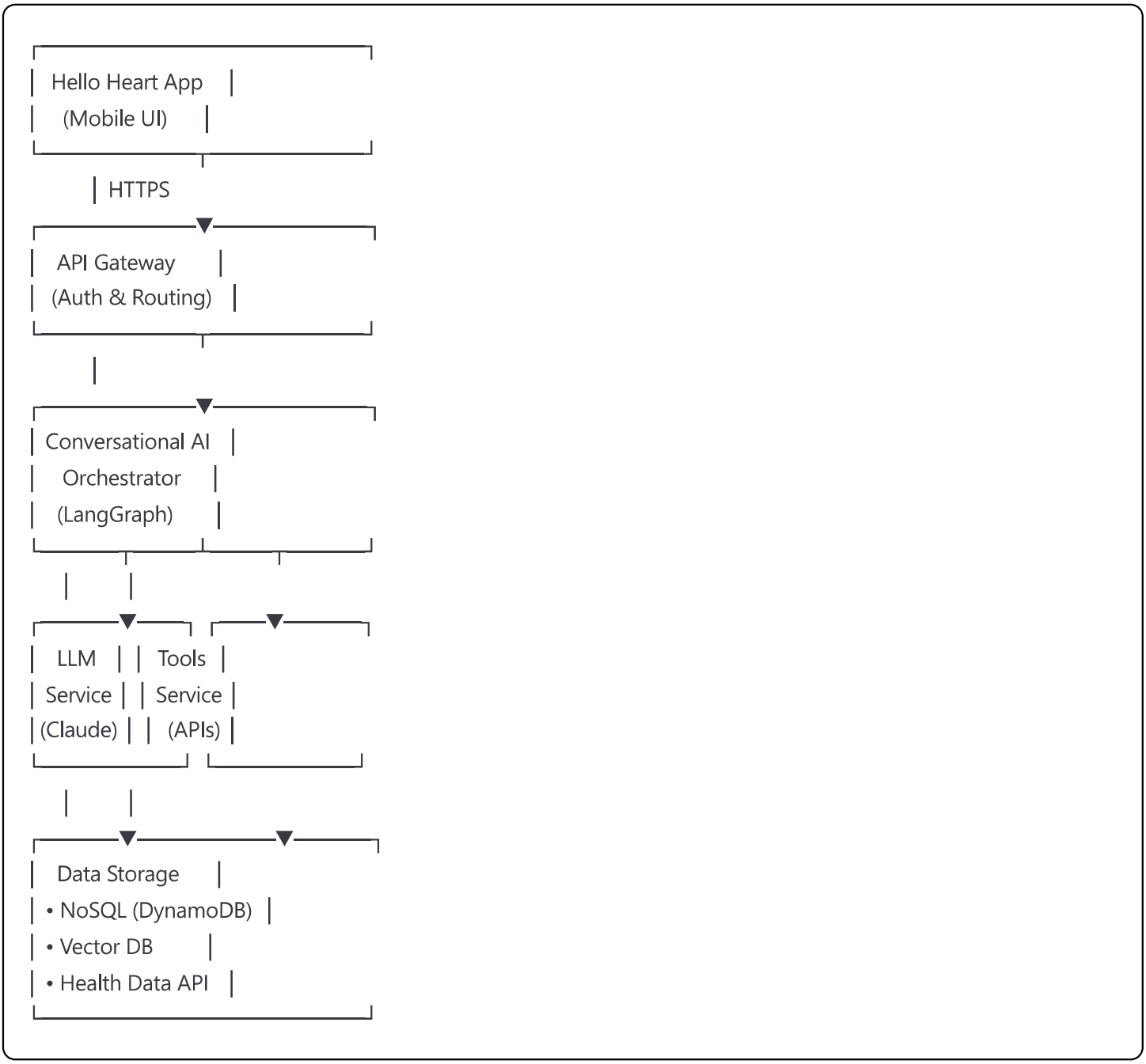
✅ **High-Level Architecture**: GenAI agents-based system with microservices
✅ **LLM Orchestration Framework**: LangGraph (with LangChain comparison)
✅ **Data Storage Strategy**: Multi-tier approach for PRD inputs and conversations
✅ **Prompt Strategy & Agent Behavior**: Dynamic prompting with safety guardrails
✅ **Production Evaluation & Monitoring**: Comprehensive metrics and feedback loops

1. High-Level Architecture

System Overview

The conversational AI assistant integrates with the Hello Heart mobile application as a cloud-native microservice, leveraging modern **GenAI agents** architecture to provide personalized health insights and proactive engagement. The system employs a **multi-agent pattern** where specialized agents handle different aspects of health monitoring and user interaction.

Architecture Components



Key Design Principles

- **Event-Driven Architecture**: Supports real-time health data integration and proactive nudges
- **Microservices Pattern**: Separates concerns for scalability and maintainability
- **Serverless-First**: Leverages AWS Lambda/Cloud Functions for automatic scaling
- **Multi-Modal Ready**: Architecture supports future voice/image inputs

2. LLM Orchestration Framework

LangGraph Selection Rationale

We choose **LangGraph** over LangChain for its superior handling of:

- **Stateful Conversations**: Explicit state management for complex health dialogues
- **Conditional Flows**: Support for medical decision trees and guided interactions
- **Cyclic Workflows**: Enables follow-up questions and iterative health assessments
- **Multi-Agent Coordination**: Native support for orchestrating multiple specialized agents

Framework Comparison

Feature	LangChain	LangGraph	Decision
State Management	Implicit	Explicit	✅ LangGraph
Cyclic Workflows	Limited	Native	✅ LangGraph
Agent Orchestration	Basic	Advanced	✅ LangGraph
Production Readiness	Good	Excellent	✅ LangGraph

Agent Architecture

```python
# Simplified LangGraph State Definition
class HealthAssistantState(TypedDict):
    messages: List[Message]
    user_health_data: Dict
    current_intent: str
    requires_medical_disclaimer: bool
    conversation_phase: Literal["greeting", "assessment", "advice", "follow_up"]
```

Agent Nodes

1. **Intent Recognition Node**: Classifies user queries (health query, emergency, out-of-scope)
2. **Data Retrieval Node**: Fetches relevant health metrics from user history
3. **LLM Response Node**: Generates personalized insights using Claude
4. **Safety Check Node**: Validates responses against medical guidelines
5. **Follow-up Node**: Determines if proactive nudges are needed

3. Data Storage Strategy

Multi-Tier Storage Architecture

1. Conversation History (DynamoDB)

```yaml
```

Table: ConversationHistory
PartitionKey: USER#<*userId*>
SortKey: CONV#<*conversationId*>#MSG#<*timestamp*>
Attributes:
- content: String
- role: Enum[user, assistant]
- metadata: JSON
- ttl: Number (30 days retention)

2. Health Metrics (Time-Series)

- Real-time data ingestion via Kinesis/Kafka
- Storage in InfluxDB or TimeStream for efficient queries
- Aggregated daily/weekly summaries in DynamoDB

3. Knowledge Base (Vector Database)

- **Pinecone/Weaviate** for semantic search
- Stores medical guidelines, FAQs, and educational content
- Enables RAG (Retrieval Augmented Generation) for accurate responses

Data Privacy & Security

- End-to-end encryption for PHI (Protected Health Information)
- HIPAA-compliant infrastructure
- Data anonymization for analytics
- User consent management for data usage

4. Prompt Strategy & Agent Behavior

Persona Definition

You are a compassionate, evidence-based health coach for Hello Heart.
Your role is to:
- Provide personalized insights based on user health data
- Offer actionable advice in simple, encouraging language
- Maintain a positive, supportive tone
- Never diagnose or prescribe medication
- Redirect medical emergencies to appropriate care

Dynamic Prompt Construction

```python
def compose_prompt(user_query: str, health_context: Dict) -> str:
    return f"""
    [System Instructions]
    {PERSONA_PROMPT}

    [User Health Context]
    Recent BP: {health_context['blood_pressure']}
    Weekly Steps: {health_context['step_count']}
    Sleep Quality: {health_context['sleep_score']}
    Heart Rate Variability: {health_context['hrv']}

    [Conversation History]
    {format_recent_messages(last_n=3)}

    [Current Query]
    User: {user_query}

    [Response Guidelines]
    - Start with direct answer
    - Include relevant data insights
    - End with actionable suggestion or follow-up question
    - Maintain encouraging, supportive tone
    """
```

Agent Design Principles

1. **Modularity**: Each agent handles specific health domains (BP, activity, sleep)
2. **Composability**: Agents can be combined for complex health assessments
3. **Interpretability**: Clear reasoning paths for all health recommendations
4. **Safety-First**: Multiple validation layers before delivering advice

Behavioral Guardrails

- **Medical Disclaimer Triggers**: Automatic disclaimers for symptom-related queries
- **Emergency Detection**: Immediate redirection for critical symptoms (chest pain, severe BP)
- **Scope Boundaries**: Polite deflection for non-health queries
- **Prompt Injection Defense**: Input validation and response filtering

5. Production Evaluation & Monitoring

Comprehensive Metrics Framework

Performance Metrics (Real-time)

- **Response Latency**: p50 < 1s, p95 < 2s, p99 < 3s
- **Throughput**: 1000+ requests/minute capacity
- **API Availability**: 99.9% uptime SLA
- **Token Efficiency**: <500 tokens/response average

Quality Metrics (Daily Analysis)

- **User Satisfaction**: In-app ratings (target: 4.5+/5.0)
- **Conversation Completion**: >80% reach natural conclusion
- **Follow-up Engagement**: >60% respond to proactive nudges
- **Health Outcome Correlation**: Track BP/activity improvements

Business Metrics (Weekly Review)

- **Cost per Conversation**: Target <$0.10
- **User Retention**: 30-day retention >70%
- **Feature Adoption**: New feature usage within 7 days
- **Clinical Accuracy**: Expert review score >95%

Advanced Logging Architecture

```yaml
Logging Strategy:
  Structured Logs:
    - Request/Response pairs with correlation IDs
    - User interactions with anonymized PII
    - System performance metrics
    - Error traces with full context

  Log Destinations:
    - CloudWatch Logs: Real-time analysis
    - S3: Long-term storage and ML training
    - ElasticSearch: Full-text search capability
    - Datadog: Custom dashboards and alerting
```

Feedback Mechanisms

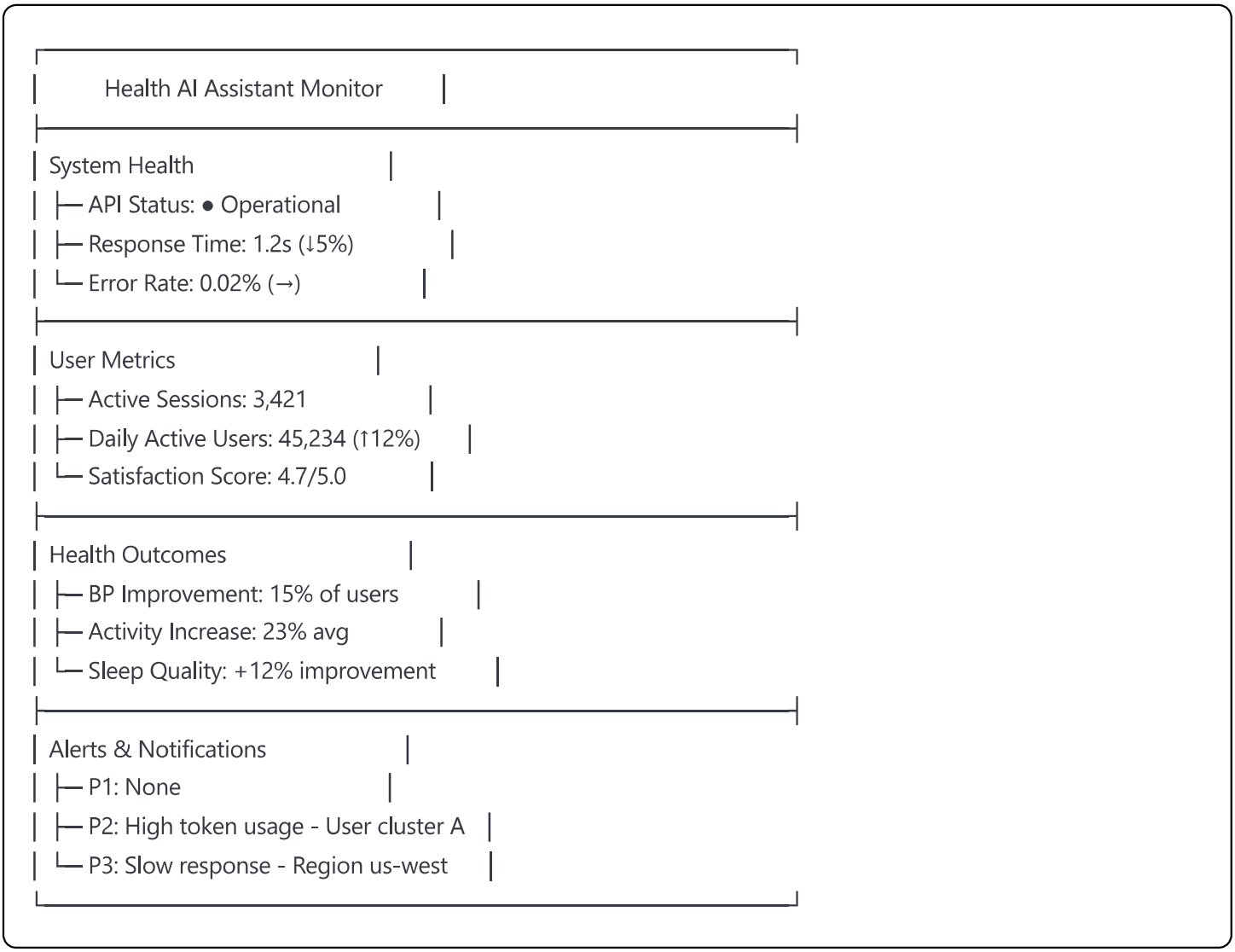
User Feedback Collection

```python
class FeedbackCollector:
    def collect_feedback(self, conversation_id: str):
        return {
            "satisfaction_rating": 1-5,
            "helpfulness_score": 1-10,
            "would_recommend": boolean,
            "improvement_suggestions": text,
            "health_goal_progress": percentage
        }
```

Clinical Review Process

1. **Automated Flagging**: AI identifies conversations needing review
2. **Expert Queue**: Clinical team reviews flagged interactions
3. **Feedback Loop**: Improvements fed back to prompt engineering
4. **Compliance Tracking**: Ensure medical guideline adherence

Real-time Monitoring Dashboard



Implementation Roadmap

Phase 1 (Months 1-2): Core Infrastructure

- Set up LangGraph orchestration
- Integrate Claude API
- Implement basic health data retrieval

Phase 2 (Months 3-4): Safety & Scale

- Add medical guardrails
- Implement comprehensive monitoring
- Load testing for 100+ concurrent users

Phase 3 (Months 5-6): Intelligence & Optimization

- Deploy RAG for knowledge base
- Implement proactive nudges

- A/B testing framework

Future Considerations

- Multi-modal inputs (voice, images)
- Integration with wearable devices
- Predictive health insights using ML
- Clinical trial participation