# High Level Design Document

## Introduction

This High Level Design (HLD) document outlines the architecture and core components for **FinScenarioMap - AI-Powered Banking Scenario Mapping Tool**. The system enables users to input or upload hypothetical banking scenarios, which are then mapped to similar historical cases using AI, with actionable recommendations generated for each scenario. The design emphasizes modularity, security, and scalability, leveraging Python, Langgraph, agentic-ai, generative-ai, and OpenAI.

---

## 1. System Architecture Overview

**Architecture Summary:**
The system is structured as a modular workflow, orchestrated by Langgraph, integrating AI-driven scenario analysis and recommendations. User interactions are handled via a web/API interface, with backend services managing scenario processing, historical mapping, and recommendation generation.

| Module | Role |
|---|---|
| User Interface/API | Accepts scenario input/upload, displays results |
| Scenario Processor | Validates and preprocesses user scenarios |
| Workflow Orchestrator | Manages scenario mapping workflow (Langgraph) |
| Historical Mapper | Matches input scenarios to historical cases (agentic-ai, generative-ai) |
| Recommendation Engine | Generates actionable insights (OpenAI, generative-ai) |
| Data Store | Stores historical cases, scenarios, and results |

---

## 2. Component Interactions

| Step | Interaction |
|---|---|
| 1 | User submits scenario via UI/API |
| 2 | Scenario Processor validates and preprocesses input |
| 3 | Workflow Orchestrator (Langgraph) initiates mapping workflow |
| 4 | Historical Mapper finds similar historical cases using agentic-ai/generative-ai |
| 5 | Recommendation Engine generates actions using OpenAI/generative-ai |
| 6 | Results are stored in Data Store and returned to User Interface/API |

---

## 3. Data Flow Overview

| Source | Data | Destination | Purpose |
|---|---|---|---|

| | | | |
|---|---|---|---|
| User | Scenario input/file | Scenario Processor | Input validation and preprocessing |
| Scenario Processor | Processed scenario | Workflow Orchestrator | Initiate mapping workflow |
| Workflow Orchestrator | Scenario context | Historical Mapper | Find similar historical cases |
| Historical Mapper | Matched cases | Recommendation Engine | Generate recommendations |
| Recommendation Engine | Recommendations | Data Store, UI/API | Persist and present results |

---

## 4. Technology Stack

| Component | Technology/Framework |
|---|---|
| Backend | Python |
| Workflow Orchestration | Langgraph |
| AI/ML | agentic-ai, generative-ai |
| LLM Integration | OpenAI |
| Data Storage | Relational/NoSQL DB |
| API/UI | REST API, Web Framework |

---

## 5. Scalability & Reliability

- **Modular Design:** Each component is independently deployable and replaceable.
- **Scalability:** Stateless processing enables horizontal scaling of scenario processing and AI components.
- **Security:** User data is validated and securely stored; AI interactions are sandboxed.
- **Reliability:** Workflow orchestration ensures fault tolerance and traceability of scenario processing.

---

**End of Document**