Software Requirement Specification (SRS)

Project Name: Enigma

Version: 1.0

Date: November 24, 2025

Type: AI-Powered Academic Research Ecosystem

---

1. Executive Summary

Enigma is an AI-powered research platform designed to transcend standard chatbots. Unlike tools that simply "answer questions," Enigma functions as a **long-term research partner**. It assists students and academics through the entire lifecycle of a project: from initial literature review and methodology selection to data analysis (coding) and final drafting.

The system is built on a **Modular Serverless Architecture** to ensure infinite scalability and low maintenance costs, utilizing a "Hybrid Context" strategy to manage API costs effectively.

---

2. System Architecture

2.1 High-Level Design

The system follows a modular, event-driven serverless pattern.

- **Frontend Layer: Firebase** (Auth, Hosting, Firestore). Handles user interaction and real-time state updates.
- **Orchestration Layer: n8n**. Acts as the API Gateway and logic router, managing workflows between the frontend and AI services.
- **Cognitive Layer (The Brain): Google Cloud Run** hosting Python containers running **LangGraph**. Handles complex, multi-step agentic reasoning.
- **Execution Layer (The Hands): E2B**. A secure sandboxed code interpreter for running Python data analysis and generating visualizations.
- **Data Layer (The Memory):**
  - **Supabase (PostgreSQL):** Relational data for Projects, Citations, and Artifacts.
  - **Pinecone:** Vector storage for semantic search.
  - **Firestore:** Real-time session chat logs.

Refer to the architectural diagram (Fig 1) for visualization.

---

3. User Modes (Functional Modules)

Enigma operates in three distinct modes based on user intent.

Module A: The Tutor (Interactive Study)

Target Audience: Students preparing for exams or understanding textbooks.

- **FR-A1 (Ingestion):** The system shall use **LlamaParse** to extract text, tables, and formulas from PDF/PPTX uploads, converting them to clean Markdown.
- **FR-A2 (Context Caching):** For single-session document chats, the system shall utilize **Gemini 1.5 Pro Context Caching** to reduce token costs by ~95% for repeated queries on large books.
- **FR-A3 (Active Recall):** The system shall act as a Socratic Tutor. Instead of summarizing, it shall generate JSON-structured Flashcards and Quizzes upon request.

Module B: Deep Research (The Analyst)

Target Audience: Users needing a one-off, comprehensive report.

- **FR-B1 (Multi-Agent Loop):** The system shall employ a **LangGraph** architecture with the following agents:
  - *Planner:* Deconstructs topics into sub-queries.
  - *Researcher:* Uses **Tavily/SerpAPI** to scrape web data.
  - *Reviewer:* Critiques findings and triggers loops if data is insufficient.
  - *Writer:* Synthesizes the final report.
- **FR-B2 (Async Processing):** To prevent browser timeouts, deep research tasks shall run asynchronously via **Firebase Cloud Functions** triggering **n8n**, updating the UI via Firestore listeners when complete.

Module C: The Research Partner (Project Workspace)

Target Audience: Thesis students and Researchers (Long-term).

- **FR-C1 (Project State):** The system shall persist the state of a research project (Ideation $\to$ Methodology $\to$ Analysis) using **Supabase**.
- **FR-C2 (Citation Vault):** The system must store every valid source found in a dedicated `citations` database table. The Writer Agent is strictly constrained to cite only sources existing in this table.
- **FR-C3 (Methodology Advisor):** The system shall analyze user requirements and suggest research methods (e.g., "Use ARIMA for this time-series data") based on uploaded literature.
- **FR-C4 (Data Sandbox):** The system shall allow users to upload datasets (CSV/Excel). The AI shall write and execute Python code in **E2B** to generate PNG graphs and statistical insights.

---

4. Data Requirements (Schema Strategy)

The system requires a hybrid database approach to handle structured relationships and unstructured chat.

4.1 Relational Schema (Supabase/PostgreSQL)

To ensure citation integrity and project management:

- `Projects`: Stores metadata and current phase (Lit Review, Analysis, etc.).
- `Citations`: Stores Title, Author, URL, Abstract, and **Vector Embeddings**.
- `Artifacts`: Stores URLs to generated graphs and code scripts.
- `Citation_Links`: A join table linking specific Chat Messages to specific Citation IDs (for footnotes).

Refer to Entity Relationship Diagram (Fig 2) for detailed visualization.

4.2 Vector Schema (Pinecone)

- Used for "Library Search" mode.
- Chunks: 1000 tokens with 200 token overlap.
- Embedding Model: **Voyage-law-2** or **OpenAI text-embedding-3-small**.

---

5. Non-Functional Requirements

5.1 The "Economic Shield" (Cost Control)

- **NFR-1 (Model Routing):** The system shall default 90% of interactions to **Gemini 1.5 Flash**. Routing to **GPT-4o** or **Gemini 1.5 Pro** is reserved for complex reasoning or "Pro" tier users.
- **NFR-2 (Semantic Caching):** Before calling an LLM, the system shall check the Vector DB for semantically identical previous questions. If a match (>95% similarity) is found, the cached answer is returned ($0 cost).
- **NFR-3 (Input Cleaning):** The system shall run a regex script to strip headers, footers, and bibliographies from PDFs before tokenization.

5.2 Performance & Reliability

- **NFR-4:** Chat responses must stream within 2 seconds.
- **NFR-5:** Deep Research reports must provide intermediate status updates (e.g., "Scanning web...", "Drafting...") to the UI to maintain user engagement during long wait times (1-3 mins).

---

6. Technical Stack Summary

Component	Technology Selection	Justification
Frontend	React / Firebase	Fast iteration, built-in Auth, Real-time DB.
Orchestrator	n8n	Low-code backend logic, easy API integration.
AI Runtime	Python (FastAPI) on Cloud Run	Required for LangGraph state loops.
Agent Framework	LangGraph	Enables cyclic, self-correcting agent flows.
Code Sandbox	E2B	Secure execution of generated Python code.
PDF Parser	LlamaParse	Best-in-class for academic formatting (tables/math).
Database	Supabase (PostgreSQL)	Strong relational integrity for citations.

---

7. Development Roadmap

Phase 1: The MVP (Weeks 1-3)

- **Goal:** "The Tutor" (Chat with PDF).
- **Stack:** Firebase + n8n + Gemini Flash.
- **Deliverable:** Users can upload a PDF, chat with it, and generate flashcards.

Phase 2: Intelligence Layer (Weeks 4-6)

- **Goal:** "Deep Research" (The Perplexity Killer).
- **Stack:** Add Cloud Run + LangGraph.
- **Deliverable:** Users can ask a broad question and receive a cited, 2-page report.

Phase 3: The Workspace (Weeks 7-12)

- **Goal:** "The Research Partner" (Long-term projects).
- **Stack:** Add Supabase + E2B.
- **Deliverable:** Project dashboard, Citation manager, Data analysis sandbox, and Split-screen UI.

---

8. Conclusion

Enigma represents a shift from "AI as a Tool" to "AI as a Collaborator." By architecting for **state persistence** (remembering the project context) and **execution** (running code), we solve the two biggest frustrations researchers have with current LLMs: hallucination and superficiality.

9. Diagrams

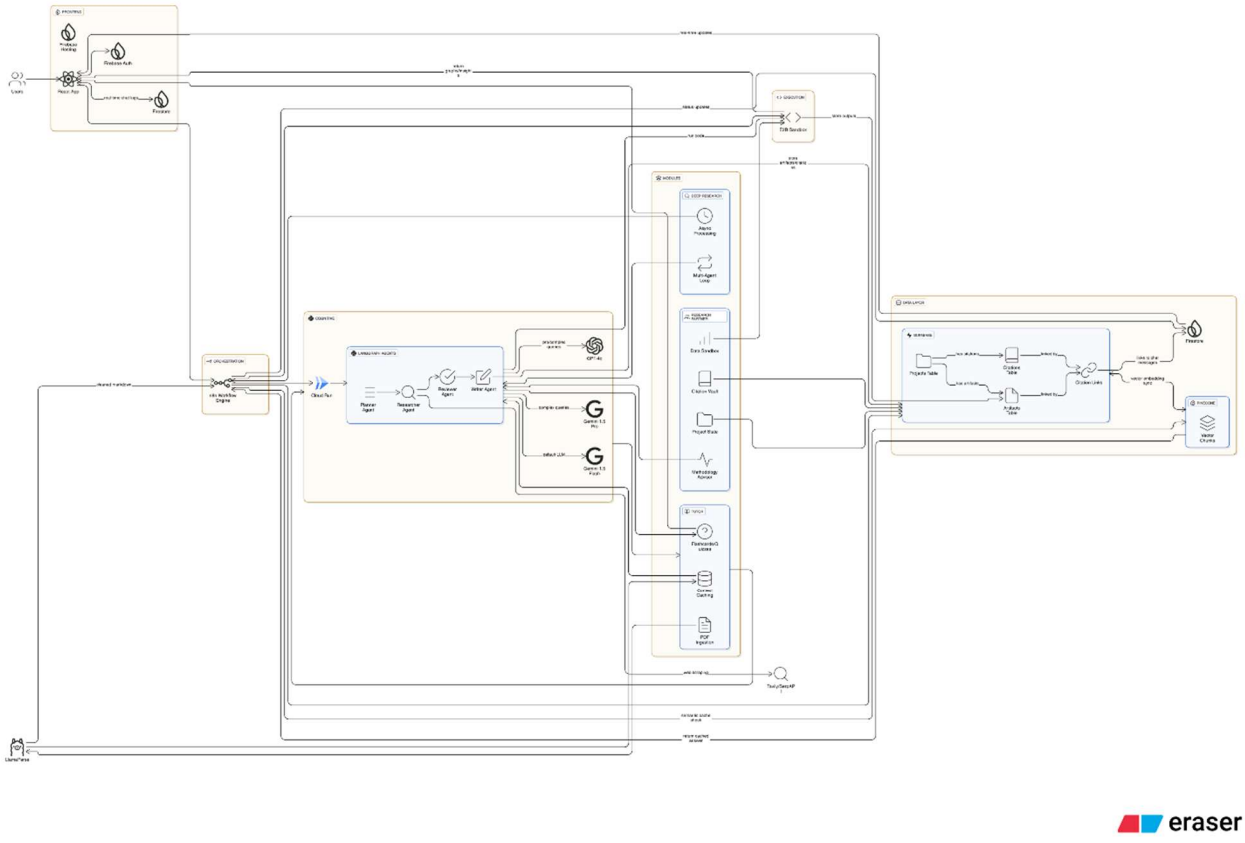

Figure 1. Architectural diagram for Enigma

Figure 2. Database Schema for Enigma Superbase Data Layer