Requirement Analysis Document (RAD)

1. Vision

The MiniRAG Chatbot is a modular, command-line based Retrieval-Augmented Generation system designed to answer inquiries regarding the Marmara University Computer Engineering department, Faculty of Engineering, and general university policies. Unlike traditional chatbots reliant on external APIs, this system operates on a **local, ethical corpus** of documents using deterministic algorithms. The primary focus is to demonstrate high-quality Object-Oriented Analysis and Design (OOAD) using GRASP and SOLID principles, ensuring traceability from use cases to code.

2. Iteration 1 Goals

- **Core Pipeline:** Implement a sequential RAG pipeline (Intent Detection -> Query Writing -> Retrieval -> Reranking -> Answer Generation) without concurrency.
- **Determinism:** Ensure that identical inputs and configurations yield identical outputs and citations, replacing real LLMs/Vector DBs with deterministic stubs.
- **Design Patterns:** Apply **Strategy** for swappable logic (e.g., switching Rerankers), **Template Method** for the pipeline orchestration, and **Observer** for tracing execution logs.
- **Data Management:** Store data solely in local JSON/YAML files (no GUI or Database).

3. Non-Functional Requirements (NFRs)

- **Modularity (SOLID):** The system must adhere to the Open/Closed Principle; adding a new Reranker or IntentDetector should happen via configuration without modifying the RagOrchestrator code.
- **Traceability:** Every main class and method must be traceable to a specific use case step.
- **Testability:** The logic must be verifiable through at least 26 unit tests covering rules, stopwords, and ranking mathematics.
- **Configurability:** The system behavior (algorithms used, constants) must be controlled entirely via a config.yaml file.
- **Reproducibility:** Execution steps must be logged to a JSONL trace file for debugging and analysis.

4. Risks

- **Data Quality:** Since document conversion to text is manual, poor formatting or encoding issues may degrade retrieval accuracy using simple keyword matching.
- **Heuristic Limitations:** Without real semantic understanding (LLMs), the rule-based IntentDetector and keyword Retriever may fail on complex or ambiguous user queries.

5. Glossary

- **Chunk:** A fixed-size text segment (300-600 tokens) derived from a source document, acting as the atomic unit of retrieval.
- **Orchestrator:** The GRASP Controller that coordinates the flow of data between pipeline stages.
- **Strategy Pattern:** A behavioral design pattern enabling the selection of algorithms (e.g., SimpleRerankervs. NoOpReranker) at runtime via configuration.
- **Intent:** The classification of what the user wants (e.g., StaffLookup, PolicyFAQ).
- **Booster:** A mechanism to inject domain-specific terms (e.g., "office", "email") into a query when a specific Intent is detected.