Marmara University Faculty of Engineering

Fall 2025 – CSE3063

Group Number: 12

Object Oriented Software Design

Requirement Analysis Document(RAD)

1. Vision and Goals

The objective of this project is to develop a modular, local Retrieval-Augmented Generation (RAG) chatbot tailored for the Marmara University Computer Engineering Department. The system will answer student inquiries regarding curriculum, staff, and university regulations by retrieving information from a curated set of local text documents.

Primary Goals:

- To implement a deterministic pipeline (Intent Detection $\rightarrow$ Retrieval $\rightarrow$ Reranking) without using external databases or live APIs[222].
- To demonstrate high-quality Object-Oriented Design using **GRASP** and **SOLID** principles, ensuring the system is testable and extensible[3333].
- To ensure full traceability of the decision-making process via JSONL logs[4].

2. Functional Requirements

- **Data Processing:** The system must ingest raw text files, normalize the content (lowercase, distinct handling), and split them into fixed-size **chunks** (100-150 words) with overlap[5].
- **Indexing:** A persistent **Keyword Index** (Inverted Index) must be generated in JSON format, mapping tokens to document chunks[6].
- **Query Processing:** The system must detect the user's **Intent** (e.g., *StaffLookup, PolicyFAQ*) based on keyword rules defined in a configuration file[7].
- **Retrieval & Ranking:** The system must retrieve relevant chunks based on Term Frequency (TF) and rerank them using deterministic scoring (proximity and title bonuses)[888].
- **Output Generation:** The system must generate a final answer citing the source document and specific chunk ID[9].

3. Non-Functional Requirements (NFRs)

- **Determinism:** Given the same input configuration and query, the system must produce the exact same output and ranking order every time[10].
- **Traceability:** Every step of the pipeline (input, output, duration) must be logged to a **JSONL** trace file using the Observer pattern[11111111].
- **Modularity:** The system must allow swapping algorithms (e.g., changing Rerankers) via the configuration file (YAML) without modifying the source code (Open/Closed Principle)[12].
- **Independence:** The system must run offline; no connection to external vector databases or LLM APIs is permitted for the baseline logic[13].

4. Risk Analysis

- **Data Integrity:** As documents are manually converted to text, formatting errors (broken sentences, missing headers) may degrade retrieval accuracy[14].
- **Morphological Limitations:** The baseline "Keyword Search" may struggle with Turkish agglutinative morphology (suffixes), failing to match queries like "dersler" with "ders" if stemming is not robust.
- **Query Ambiguity:** Short or vague user queries may lead to incorrect Intent Detection due to the limitations of rule-based logic.

5. Glossary

- **Chunk:** A specific segment of text from a source document, used as the basic unit of retrieval[15].
- **Inverted Index:** A data structure mapping unique tokens (words) to the list of chunks where they appear[16].
- **TF (Term Frequency):** The number of times a query term appears in a specific chunk[17].
- **Intent:** The category of the user's need (e.g., looking for a person vs. looking for a rule)[18].
- **JSONL:** A file format where each line is a valid JSON object, used here for logging traces[19].