

**TOBB ETU**

**Economy & Technology University**

**BIL 495 / YAP 495**

**Software Requirements Specification (SRS)**

***Reference:*** *IEEE 830-1998 / ISO/IEC/IEEE 29148:2018*

## Document Control Table

|  |  |
| --- | --- |
| Document Title | Project Closure Report |
| Document ID | (To be assigned) |
| Prepared By (Author) | Emre Ekşi |
| Reviewed By | Ahmet Babagil |
| Approved By | Ahmet Babagil |
| Preparation Date | 19/01/26 |
| Approval Date | 19/01/26 |
| Version / Revision | 1.0 |
| Confidentiality Level | Public |
| Reference Standards | IEEE 15288, IEEE 12207, INCOSE SE Handbook v5 |

## Change Record (Revision History Table)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Prepared By | Reviewed/Approved By | Description of Change |
| 1.0 | (Date) | (Author) | (Approver) | Initial Release |
| 1.1 |  |  |  | Minor editorial updates |
| 1.2 |  |  |  | Added stakeholder feedback section |

[1. Introduction](#_heading=)

[1.1 Purpose](#_heading=)

[1.2 Scope](#_heading=)

[1.3 Definitions, Acronyms, Abbreviations](#_heading=)

[1.4 References](#_heading=)

[1.5 Overview](#_heading=)

[2. Overall Description](#_heading=)

[2.1 Product Perspective](#_heading=)

[2.2 Product Functions](#_heading=)

[2.3 User Characteristics](#_heading=)

[2.4 Constraints](#_heading=)

[2.5 Assumptions and Dependencies](#_heading=)

[3. Specific Requirements](#_heading=)

[3.1 Functional Requirements](#_heading=)

[3.2 External Interface Requirements](#_heading=)

[3.3 Performance Requirements](#_heading=)

[3.4 Design Constraints](#_heading=)

[3.5 Software System Attributes](#_heading=)

[3.6 Other Requirements](#_heading=)

[4. Appendices](#_heading=)

[4. References](#_heading=h.a6qvajnlk889)

**1. Introduction**

**1.1 Purpose**

**The purpose of this document is to provide a comprehensive description of the software requirements for the AI-Based Adaptive Course Generator. It is intended for project developers, academic supervisors, and stakeholders involved in the BIL 496 graduation project.**

**1.2 Scope**

**The system is an AI-driven platform designed to generate modular educational content, including lessons, quizzes, and final exams, tailored to individual learners. It utilizes Large Language Models (LLMs) to transform user-defined topics or uploaded content into structured micro-learning units. The scope excludes Learning Management System (LMS) integration and video generation in the initial version.**

**1.3 Definitions, Acronyms, Abbreviations**

* **LLM: Large Language Model.**
* **API: Application Programming Interface.**
* **UI: User Interface.**
* **EMA: Exponential Moving Average (used for mastery scoring).**
* **RAG: Retrieval-Augmented Generation.**

**1.4 References**

* **[1] Y. Bengio et al., “Deep Learning for Adaptive Educational Systems,” Nature Machine Intelligence, 2020.**
* **[2] OpenAI, “GPT-4 Technical Report,” 2023.**
* **[3] ISO/IEC 25010 (System & Software Quality Models).**

**1.5 Overview**

**This document specifies the functional requirements, system architecture, and quality attributes necessary for the successful delivery of the adaptive learning prototype.**

**2. Overall Description**

**2.1 Product Perspective**

**This is a standalone web-based tool developed from scratch. It integrates with third-party APIs for content generation and uses open-source repositories (OpenStax, Wikipedia) for dataset support.**

**2.2 Product Functions**

* **Generate structured lessons from a topic or uploaded content.**
* **Produce multiple-choice quizzes and chapter exams.**
* **Store session-based feedback for content personalization.**
* **Deliver content through an intuitive web interface.**

**2.3 User Characteristics**

* **University Students: Seeking modular learning resources for specific topics.**
* **Independent Learners: Individuals needing structured guidance without technical backgrounds.**
* **Instructors: Educators seeking efficient ways to generate tailored course materials.**

**2.4 Constraints**

* **Economic: API usage is subject to rate limits and token costs.**
* **Legal: The system must maintain GDPR/KVKK compliance by avoiding the collection of personally identifiable information (PII).**
* **Environmental: Energy efficiency is prioritized through SHA-256 prompt caching, which reduces redundant API calls.**
* **Technical: The initial version is limited to English-only content and focused primarily on STEM subjects.**

**2.5 Assumptions and Dependencies**

* **Users have stable internet access and a modern web browser.**
* **Access to external APIs (OpenAI) remains available.**

**3. Specific Requirements**

**3.1 Functional Requirements**

* **REQ-1: The system shall allow users to input a topic name or upload text.**
* **REQ-2: The system shall generate structured lessons based on the input.**
* **REQ-3: The system shall generate quizzes and chapter-end exams.**
* **REQ-4: The system shall collect user feedback and adapt future content based on performance.**

**3.2 External Interface Requirements**

* **User Interfaces: Streamlit-based responsive UI with input fields for topics and output displays for lessons.**
* **Software Interfaces: FastAPI backend integration with LLM Service Adapters and SQLite storage.**
* **Communication Interfaces: HTTPS for secure communication with external APIs.**

**3.3 Performance Requirements**

* **Response Time: Content generation should complete within 10 seconds.**
* **Efficiency: Average response time target is $\le$ 2 seconds using prompt caching.**
* **Scalability: The system must support at least 10 concurrent users.**

**3.4 Design Constraints**

* **Use of Streamlit for the frontend.**
* **Use of Python for backend logic.**
* **Integration limited to open or free APIs due to budget constraints.**

**3.5 Software System Attributes**

* **Reliability: Centralized exception handling with a target "Uncaught Exception Rate".**
* **Security: Minimal data retention with session-based storage to ensure privacy.**
* **Maintainability: Layered architecture with unit test coverage 90%.**
* **Portability: Web-based UI accessible from both desktop and mobile browsers.**

**4. Appendices**

**4.1 Traceability Matrix (Example)**

| **Requirement ID** | **Description** | **Design Component** | **Test Level** |
| --- | --- | --- | --- |
| **R1** | **Generate lesson output** | **Lesson Parser** | **Unit Test** |
| **R2** | **Generate quiz** | **Quiz Constructor** | **Integration Test** |
| **R3** | **Provide feedback** | **Feedback Handler** | **System Test** |

**4.2 Task Distribution**

* **Emre Ekşi: UI/UX Design, Prompt Engineering, Documentation.**
* **Ahmet Babagil: Testing, Debugging, and Validation.**
* **Seda Naz Dolu: Frontend Development with Streamlit.**
* **Cemil Gündüz: Backend Development and Quiz Logic.**

**4. Appendices**

**The following appendices provide detailed technical breakdowns, project management data, and risk assessments for the system.**

**A. Initial System Architecture**

**The system utilizes a layered architecture to ensure modularity and scalability:**

1. **User Interface (UI): Implemented in Streamlit for rapid deployment.**
2. **Backend API: Built with FastAPI to manage LLM calls and user requests.**
3. **Database: SQLite for storing session data and caching.**
4. **Vector Database: FAISS for semantic search on educational content.**
5. **LLM Integration: OpenAI API for dynamic content generation.**

**B. Technology Stack Justification**

| **Component** | **Technology** | **Justification** |
| --- | --- | --- |
| **Frontend** | **Streamlit** | **Quick prototyping, minimal setup, and easy deployment.** |
| **Backend** | **FastAPI** | **High performance with asynchronous support for API creation.** |
| **Database** | **SQLite** | **Lightweight and easy to manage during development phases.** |
| **Vector DB** | **FAISS** | **Efficient similarity search for embedding-based retrieval.** |
| **AI Model** | **OpenAI API** | **High-quality text generation and complex comprehension capabilities.** |

**C. Risk Assessment Matrix**

| **Risk** | **Probability** | **Impact** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| **API Cost Overrun** | **Medium** | **High** | **Monitor usage and set budget alerts.** |
| **Model Hallucination** | **Medium** | **Medium** | **Include verification and fact-checking steps.** |
| **FAISS Indexing Errors** | **Low** | **Medium** | **Validate embeddings during the preprocessing stage.** |

**D. Roles and Responsibilities**

| **Member** | **Responsibilities** |
| --- | --- |
| **Emre Ekşi** | **Backend API, LLM integration, and FAISS setup.** |
| **Ahmet Babagil** | **Frontend development (Streamlit) and UI/UX design.** |
| **Seda Naz Dolu** | **Dataset preparation, preprocessing, and embeddings.** |
| **Cemil Gündüz** | **Testing, system evaluation, and risk management.** |

**5. References**

**The following sources were used for research and technical guidance during the project development:**

* **[1] Y. Bengio et al., “Deep Learning for Adaptive Educational Systems,” *Nature Machine Intelligence*, vol. 2, no. 8, pp. 467–476, 2020.**
* **[2] OpenAI, “GPT-4 Technical Report,” *arXiv preprint arXiv:2303.08774*, 2023. Available:.**
* **[3] J. Kaur and A. Singh, “AI in Education: Applications and Challenges,” *International Journal of Artificial Intelligence in Education*, vol. 30, no. 2, pp. 211–232, 2021.**
* **[4] A. Brown et al., “Automatic Question Generation Using Transformer Models,” in *Proc. of the 2022 Conference on Computational Linguistics*, Dublin, Ireland, pp. 1504–1516.**
* **[5] S. K. Tkalcic and L. Chen, “Adaptive Content Generation for Personalized Learning,” *ACM Transactions on Interactive Intelligent Systems*, vol. 11, no. 4, 2021.**