Technical Requirements Specification

Medical Message Classification Web Application

Document Version: 1.0
Date: May 17, 2025
Prepared by: Naumenko Daria
Conforms to: ISO/IEC/IEEE 29148:2018

---

1. Introduction

1.1 Purpose

This document specifies the technical requirements for a web-based medical message classification system designed to process user-input health-related messages and generate structured outputs and user-friendly medical reports. The requirements are defined in accordance with the **ISO/IEC/IEEE 29148:2018** standard to ensure clarity, traceability, and stakeholder alignment.

1.2 Scope

The system is a web application that enables users to submit health-related messages and receive:

- Structured output containing fields such as symptoms, category, diagnosis, recommendations, suggested medications, and generalizations.
- A detailed, patient-friendly medical report in markdown-converted HTML format.

The application integrates two fine-tuned large language models (LLMs) for processing and report generation, with a FastAPI backend, Jinja2 templating, and a modern HTML/CSS frontend. The system includes error handling, logging, and accessibility features to ensure usability and reliability.

1.3 Definitions, Acronyms, and Abbreviations

- **LLM**: Large Language Model
- **FastAPI**: A Python web framework for building APIs
- **Jinja2**: A templating engine for rendering HTML
- **JSON**: JavaScript Object Notation
- **Markdown**: A lightweight markup language for formatting text
- **UX**: User Experience
- **HTML/CSS**: HyperText Markup Language and Cascading Style Sheets

- **PEFT**: Parameter-Efficient Fine-Tuning library

1.4 References

- ISO/IEC/IEEE 29148:2018, Systems and Software Engineering – Life Cycle Processes – Requirements Engineering
- FastAPI Documentation: https://fastapi.tiangolo.com/
- Hugging Face Transformers: https://huggingface.co/docs/transformers/
- Jinja2 Documentation: https://jinja.palletsprojects.com/

---

2. Stakeholder Requirements

2.1 Stakeholder Identification

- **End Users**: Individuals submitting health-related messages, including patients, elderly people and non-technical users.
- **Developers**: Engineers maintaining and extending the system.

2.2 Stakeholder Needs

- **SHR-1**: End users shall be able to submit health-related messages through a simple web interface.
- **SHR-2**: End users shall receive a structured summary and a detailed medical report in a clear, readable format.
- **SHR-3**: The system shall be accessible on both desktop and mobile devices.
- **SHR-4**: Developers shall have access to logs for debugging and performance monitoring.
- **SHR-5**: The system shall handle errors gracefully and provide meaningful feedback to users.

---

3. System Requirements

3.1 Functional Requirements

3.1.1 User Input Processing

- **FR-1.1**: The system shall provide a web form for users to input health-related messages.
- **FR-1.2**: The system shall validate user input to ensure it is non-empty before processing.
- **FR-1.3**: The system shall submit user input to the backend via an HTTP POST request upon clicking the “Analyze and Generate Report” button.

3.1.2 Backend Processing

- **FR-2.1**: The backend shall use FastAPI to handle HTTP requests and route user input to Model 1 (Mistral).
- **FR-2.2**: Model 1 shall process the user input and return a structured JSON output containing fields: symptoms, category, diagnosis, recommendations, suggested medications, and generalization.
- **FR-2.3**: The backend shall validate the JSON output from Model 1 and handle formatting errors.
- **FR-2.4**: The backend shall transform JSON fields (e.g., convert symptom lists to comma-separated strings) for display.
- **FR-2.5**: The backend shall pass the JSON output to Model 2 (DeepSeek-R1) to generate a markdown-formatted medical report.
- **FR-2.6**: The backend shall convert the markdown report to safe HTML using custom filters.

3.1.3 Output Rendering

- **FR-3.1**: The system shall render a styled HTML table displaying the structured JSON output with formatted fields:
    - Category: Medical field and urgency level.
    - Symptoms: Comma-separated, capitalized list.
    - Suggested Medications: Pairs of medicine names and analogs.
- **FR-3.2**: The system shall render the markdown-based medical report in a styled `<div>` with preserved formatting (e.g., headings, bold text, paragraphs).
- **FR-3.3**: The system shall display both the structured table and detailed report on the same page after submission.

3.1.4 Error Handling and Logging

- **FR-4.1**: The system shall implement error handling for invalid JSON outputs from Model 1 and provide user-friendly error messages.
- **FR-4.2**: The system shall log all backend actions (e.g., input processing, model outputs, errors) for debugging and traceability.

3.2 Non-Functional Requirements

3.2.1 Usability

- **NFR-1.1**: The web interface shall use a clean, intuitive design with the Inter font and a blue-and-white color scheme.
- **NFR-1.2**: The interface shall be accessible to users without technical backgrounds.
- **NFR-1.3**: The system shall support responsive design for desktop and mobile devices.
- **NFR-1.4**: The system shall include CSS animations (e.g., fade-in effects) to enhance UX during content updates.

3.2.2 Performance

- **NFR-2.1**: The system shall process user input and render results within 3 minutes under normal conditions.

3.2.3 Reliability

- **NFR-3.1**: The system shall recover from model output errors without crashing.

3.2.4 Security

- **NFR-4.1**: The system shall convert markdown to safe HTML to prevent XSS (Cross-Site Scripting) vulnerabilities.

3.2.5 Maintainability

- **NFR-5.1**: The system shall use modular code structures for the backend (FastAPI) and frontend (Jinja2 templates).
- **NFR-5.2**: The system shall include comprehensive logging to facilitate debugging and maintenance.

3.3 System Interfaces

3.3.1 User Interface

- **SI-1.1**: The system shall provide a web-based interface with a form for text input and a submit button labeled “Analyze and Generate Report”.
- **SI-1.2**: The interface shall display results in two sections: a structured table and a detailed report `<div>`.

3.3.2 Software Interfaces

- **SI-2.1**: The backend shall interface with the Hugging Face Hub to load Model 1 (Mistral) and Model 2 (DeepSeek-R1) using the transformers and peft libraries.
- **SI-2.2**: The backend shall use Jinja2 to render HTML templates with dynamic data.

3.4 Constraints

- **CON-1**: The system shall rely on pre-trained LLMs from the Hugging Face Hub, limiting model customization to fine-tuning.
- **CON-2**: The system shall operate within the computational resources of a standard cloud server (e.g., 16GB RAM, 4 vCPUs).
- **CON-3**: The frontend shall avoid complex JavaScript frameworks to ensure lightweight performance.

---

4. Verification and Validation

4.1 Verification

- **VER-1**: Each functional requirement (FR-1.1 to FR-4.2) shall be tested through unit and integration tests to ensure correct behavior.
- **VER-2**: Non-functional requirements (e.g., performance, usability) shall be verified through user testing and load testing.

4.2 Validation

- **VAL-1**: The system shall be validated with end users to confirm that the interface is intuitive and the outputs are clear.
- **VAL-2**: Sample outputs shall be reviewed by medical professionals (if available) to ensure clinical relevance.

---

5. Appendices

5.1 Assumptions

- Users have basic internet access and a modern web browser (e.g., Chrome, Firefox).
- The Hugging Face Hub is available for model loading during operation.

5.2 Traceability Matrix

Requirement ID	Description	Test Case
SHR-1	Submit health-related messages	User submits text via form
FR-1.1	Provide web form	Form renders correctly
FR-2.2	Model 1 returns JSON	JSON contains required fields
NFR-1.1	Clean, intuitive design	User survey confirms usability

---

6. Document Control

6.1 Approval

This document requires approval by the thesis supervisor and relevant stakeholders.

6.2 Revision History

Version	Date	Description	Author
1.0	2025-05-17	Initial draft	[Your Name]