**Detailed Requirements Document:   
Real-Time Whiteboard Transcription System**

**Project Overview**

The Real-Time Whiteboard Transcription System aims to capture and transcribe whiteboard content in real time, providing an efficient tool for lecture documentation and note-taking. This includes recognizing text, mathematical symbols, and handwritten content, and converting it into a digital format. The system integrates AI-based OCR and language models to refine and structure the transcription output.

**Objectives:**

* Enable real-time transcription of whiteboard content.
* Incorporate mathematical symbol recognition and accurate handwriting analysis.
* Provide intuitive export options for processed content.
* Design a user-friendly interface tailored for students, educators, and professionals.

**Stakeholders**

The stakeholders for this project include:

* **Students and Educators:** Primary users seeking to capture and utilize lecture content efficiently.
* **Educational Institutions:** Schools and universities aiming to enhance the accessibility of educational material.
* **Software Developers:** Individuals responsible for building, maintaining, training the module and updating the system.
* **AI API Providers:** Nvidia and OpenRouter as key third-party providers.

**Functional Requirements Key Features**

**1. Real-Time Transcription:**

* Capture and transcribe whiteboard content in real time with minimal latency.

**2. Mathematical Formula Recognition:**

* Accurately detect and transcribe mathematical symbols and formulas.

**3. Handwriting Recognition:**

* Convert handwritten text into readable digital formats.

**4. Content Filtering:**

* Automatically ignore erased or irrelevant whiteboard content.

**5. Export Options:**

* Export transcribed content in formats such as .docx, .txt, and .pdf.

**6. Editing and Collaboration tool:** Allow users to edit text and add comments before exporting.

**7. Use Cases:**

* Pre-condition: User has access to the content written on a whiteboard that captures it, which can be a live-feed video or a recording.
* Basic Flow: The system allows users to upload a video, and it transcribes the content.
* Alternate Flow: The system captures and transcribes content in real time.
* Post-condition: Transcribed content is presented and available for use.

**8. Editing and Collaboration Tools:**

* Allow manual texts corrections from the user.
* User can add comments of his own to the presented content.

**Non-Functional Requirements**

**1. Performance:**

* Real-time transcription should occur with a delay of no more than 2 seconds for one frame.

**2. Usability:**

* Offer a simple and intuitive interface accessible across various devices with cross platform support.

**3. Accuracy**

* The system shall achieve at least **95% character-level accuracy** for printed and handwritten **English text** under ideal conditions (e.g., clear, high-resolution input with good lighting).
* The system shall achieve at least **85% symbol recognition accuracy** for **mathematical formulas and expressions** under ideal conditions.
* The system shall maintain a **minimum of 80% overall accuracy** when processing low-resolution images, images with visual noise, or partially occluded text.

**4. Ease of Maintenance**

* The system architecture shall support **modular updates** of OCR and mathematical recognition models without requiring a full system redeployment.

**5. Documentation**

* The system shall include a **comprehensive user manual** detailing installation, usage, and export steps, in **English**.
* A **developer guide** shall be provided, including system architecture, model update instructions, API documentation, and data format descriptions.
* All documentation must be version-controlled and updated with each major software release.

**Architectural Requirements**

**System Architecture:**

* **Frontend:** Built with React.js for video/image upload and preview.

**• Backend:** Python Flask server for OCR integration and processing.

**• APIs:** Integration with Nvidia NIM (LLaMA 4) and OpenRouter (Gemini 2.0) for AI processing.

**• Export:** Document generation using file-saver libraries.

**Performance Indicators:**

* Transcription processed within 2 seconds per frame (goal).
* Supports batch processing of pre-recorded videos.

**Technological Requirements**

**Programming Languages and Frameworks:**

* Programming Languages: Python (Flask) for backend, JavaScript (React.js) for frontend.
* AI Models: LLaMA 4 via Nvidia NIM for OCR and math recognition.
* Text Refinement: Gemini 2.0 via OpenRouter.
* Tools: FFMPEG (frame extraction), Pillow (image preprocessing),

**Example Use Case**

**Use Case Name: Transcribe Mathematical Content from Whiteboard  
Primary Actor: Student  
Stakeholders and Interests:**

* **Student: Wants accurate and fast transcription of whiteboard content for studying and sharing.**
* **Educator: Benefits from improved knowledge retention by students.**

**Preconditions:**

* **The student has access to a device (camera or smartphone) capturing the whiteboard in real time or as a recording.**
* **The system is installed and accessible to the student.**

**Basic Flow:**

1. **The student starts the system and uploads or streams the video of the whiteboard.**
2. **The system extracts frames and applies preprocessing (noise reduction, contrast enhancement).**
3. **OCR and mathematical recognition modules analyze the frames and transcribe text and symbols.**
4. **The student previews the transcribed content.**
5. **The student exports the content as a .pdf file.**

**Alternate Flows:**

* **AF1 – Manual Correction: If the student notices transcription errors, they can manually edit the transcribed content before export.**
* **AF2 – Export to Other Formats: The student can choose to export in .docx or .txt instead of .pdf.**

**Postconditions:**

* **The transcription is saved in the selected format and available for review or sharing.**

**Success Guarantee (Expected Outcome):**

* **The student receives an accurate, well-formatted digital version of the whiteboard content, including mathematical expressions.**

**Conclusion**

The Real-Time Whiteboard Transcription System bridges traditional board-based teaching with modern digital tools. It streamlines lecture documentation by leveraging advanced AI and OCR technologies, ensuring content is accessible, structured, and exportable. With modular architecture and a user-centered design, it is a versatile tool for students, educators, and professionals alike.