STEMMY Platform

.

1. Introduction

1.1 Purpose

This SRS describes the functional and non-functional requirements for **STEMMY**, an intelligent, multimodal educational platform that enhances STEM learning using natural language processing, multimodal inputs (speech, sign language, facial expression), and adaptive explanations with humor.

1.2 Scope

STEMMY will support the following subject domains: Mathematics, Science (Physics, Chemistry, Biology), and Computer Science. It accepts user inputs via typed text, speech (speech-to-text), and sign language (video input), and augments responses based on detected emotional state and user preferences. The platform will produce step-by-step explanations, simplified summaries, and humorous variants. It will include features to support learners with disabilities (deaf/hard-of-hearing, other accessibility needs).

1.3 Definitions, Acronyms and Abbreviations

- SRS: Software Requirements Specification
- RAG: Retrieval-Augmented Generation
- STT: Speech-to-text
- YOLOv11: Object detection/emotion detection model (project-specific)
- Lama 3.23b: LLM used with RAG for knowledge-grounded responses
- BERT: Transformer architecture used for classifiers/paraphraser labeling

.

2. Overall Description

2.1 Product Perspective

STEMMY is a web-based platform that interacts with several ML components and third-party services. The architecture follows a modular microservice pattern: Frontend, Backend APIs, ML/Inference services, and Data & Storage.

2.2 User Classes and Characteristics

- **Learners (General):** Students seeking explanations, step-by-step solutions, or practice. Varying ages, basic digital literacy.
- **Learners (Diverse Abilities):** Deaf/Hard-of-Hearing users who may use sign-language input; visually impaired users who rely on speech output; users with emotional needs who benefit from adaptive tone.

- **Teachers / Tutors:** Use platform to generate materials, check answers, and customize explanations.
- **Administrators / Data Engineers:** Manage datasets, pipeline, and moderation.

2.3 Operating Environment

- Web browsers supporting HTML5/ES6; mobile and desktop responsive design.
- Backend hosted on cloud (examples: AWS/Azure/GCP) with GPU instances for inference (LLM, STT).
- Persistent storage for user data, model artifacts, and logs.

2.4 Design & Implementation Constraints

- Real-time or near-real-time response targets for conversational interactions.
- Use of pre-trained models: Whisper (STT), YOLOv11 (emotion/face detection), Lama 3.23b w/ RAG, BERT classifiers, GAN for images.

.

3. Functional Requirements (FR)

1- Multimodal Input Handling

- **Description:** Accept user input via typed text, speech (microphone), and sign language (webcam video).

- **Behavior:**
    - STT service (Whisper) converts speech to text.

    - Sign-language module classifies sign inputs to textual commands (or falls back to guided sign-language keyboard).

2- Topic Classification

- **Description:** Classify incoming user requests into one of the supported subjects: Mathematics, Science, Computer Science, or request clarification.

3- Emotion Detection Input

- **Description:** Use YOLOv11 (or equivalent) on live or uploaded face/video frames to detect facial expressions and map to an emotion vector that influences tone and paraphrasing.

- **Behavior:** If no face detected or user denies camera, system falls back to explicit user-chosen mood.

4- Knowledge Retrieval & Response Generation

- **Description:** Use Lama 3.23b with RAG and curated educational datasets (18 datasets) to answer STEM queries with step-by-step explanations.

- **Behavior:**
    - Retrieve relevant documents or examples from indexed educational knowledge base.
    - Generate an explanation tailored to user level (novice/ intermediate/advanced).
    - Include optional humorous phrasing regulated by the sense-of-humor classifier.

5- Sense-of-Humor Control

- **Description:** Allow responses to include light humor; control level via user preference or automated tone mapping from detected emotion.

- **Behavior:** Use a BERT-based classifier/labeler to tag training data for humorous content; produce humor-safe content (no insults, bias, or inappropriate content).

6- Paraphrasing by Emotion or Tone

- **Description:** Paraphrase generated content to match an emotional tone (encouraging, neutral, energetic).

7- Accessibility for Deaf/Hard-of-Hearing

- **Description:** Full support for sign language input, and for speech outputs produce captions and text.

- **Behavior:** Provide a sign-language input workflow and translated text output, and allow exported transcripts.

8- User Profiles & Personalization

- **Description:** Store user preferences: preferred tone (humorous/serious), preferred subject level, saved progress, favorite explanations.

9- Admin / Content Curation Tools

- **Description:** Admin UI for dataset management, content moderation, and training pipeline triggers.

10- Analytics & Logging

- **Description:** Collect anonymized usage metrics (queries per subject, success rates, time-to-completion) and model performance telemetry.

.

4. External Interfaces

4.1 Frontend (Web UI)

- Responsive UI built with React

4.2 Backend API

- FastApi that works well for ML/AI projects

4.3 ML Services

- **STT:** Whisper-based inference API
- **Sign-language recognition:** Video-to-text classifier (custom model)
- **Face/emotion:** YOLOv11-based inference service
- **LLM:** Lama 3.23b with RAG retrieval stack
- **Classifiers:** BERT-based humor and category classifiers

.

5. Non-Functional Requirements (NFR)

1- Performance

- Average response time for text-only queries: < 2 seconds
- For multimodal queries (speech + sign): target end-to-end < 8–12 seconds depending on model loads.

2- Scalability

- Must scale horizontally for stateless services and have autoscaling for inference GPU nodes.

3- Availability

- 99.5% uptime for the web application (excluding scheduled maintenance).

4- Accessibility

- Conform to WCAG 2.1 AA accessibility standard.

.

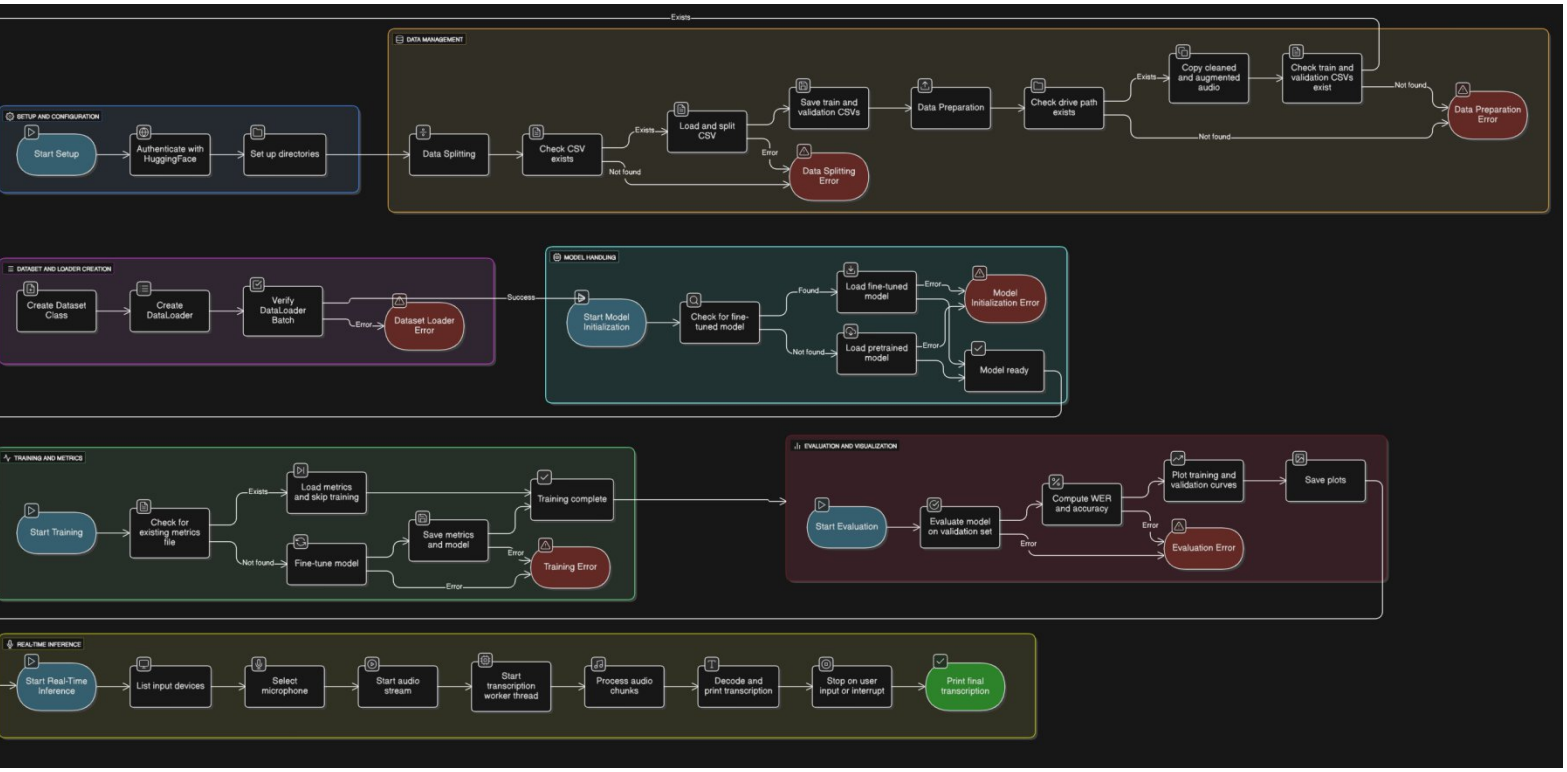
6. Glossary

- **RAG:** Retrieval-Augmented Generation
- **LLM:** Large Language Model
- **WCAG:** Web Content Accessibility Guidelines

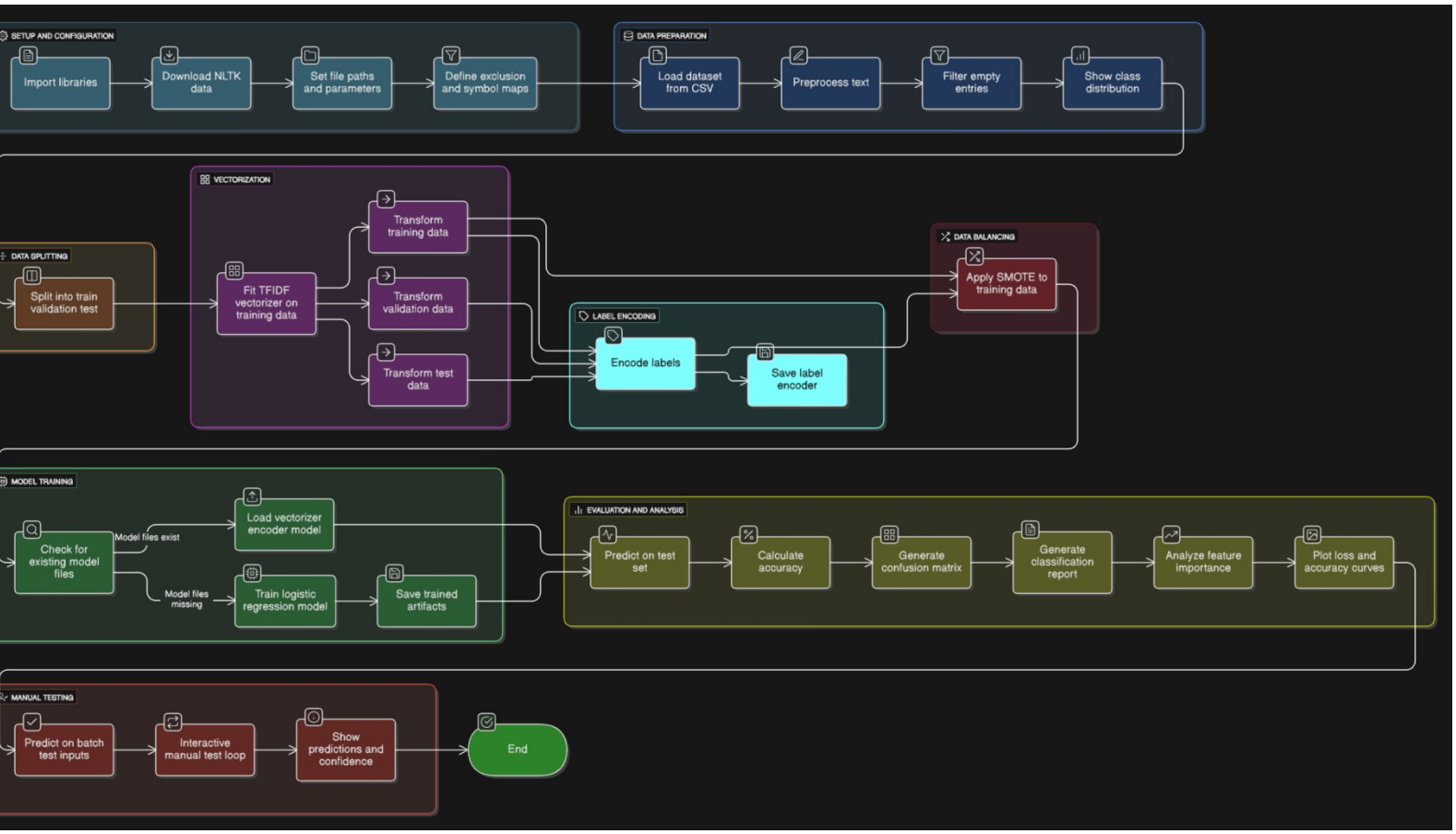
.

7. Architecture Diagrams

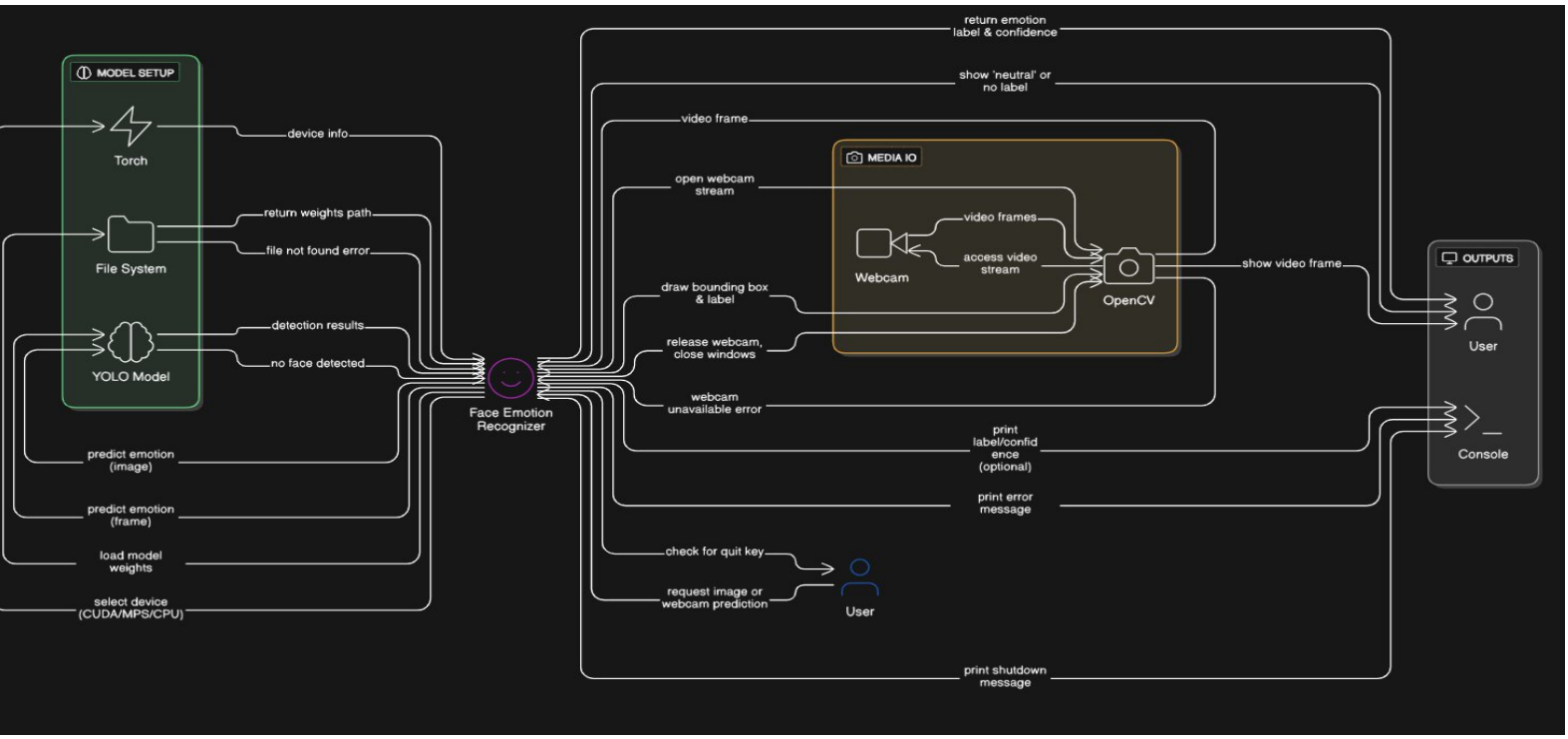
- STT:



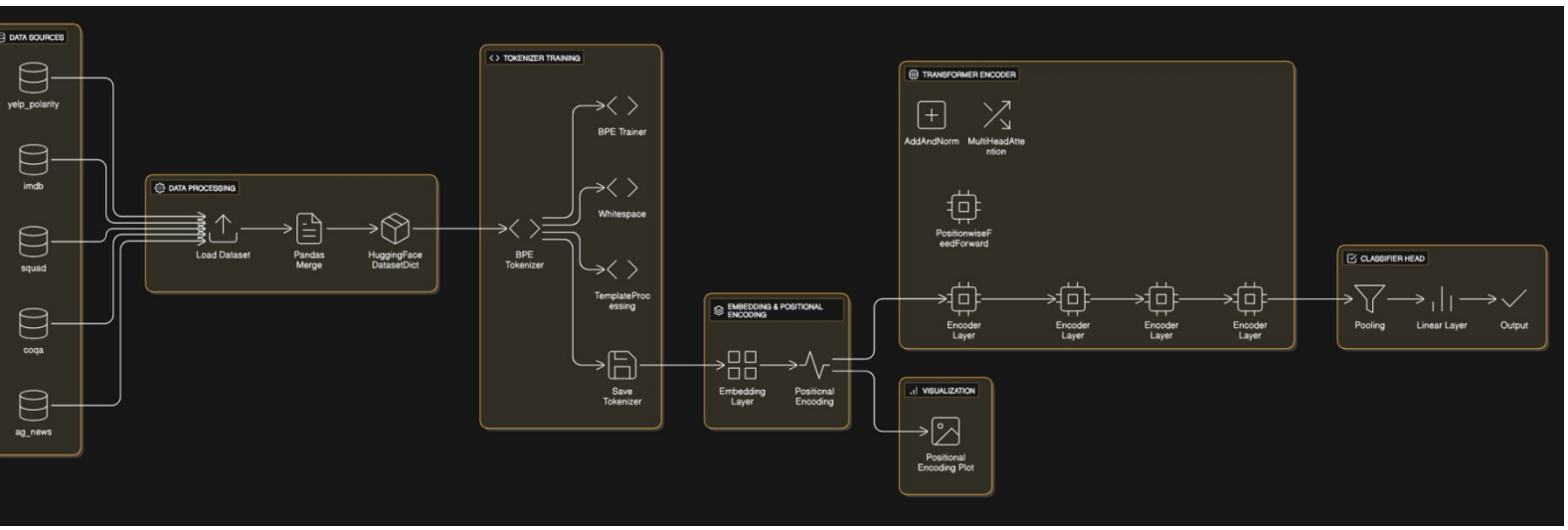
- Topic Classifier:



- YoloFace Recognition:



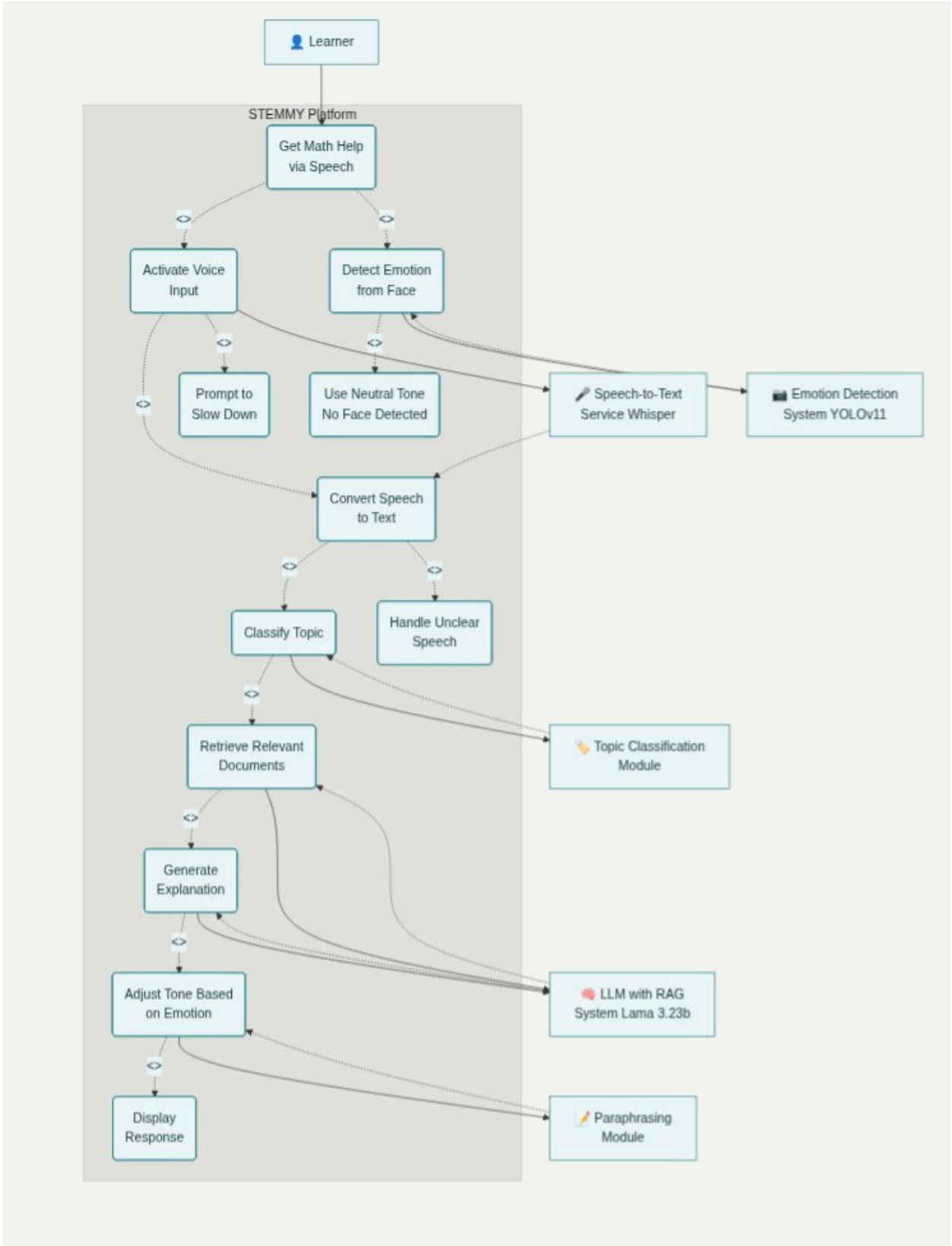
- Sense of Humor Detection:



8. use Case scenario

Use Case Name: Get Math Explanation via Speech with Emotion Recognition

Description: The Learner is audibly frustrated and asks a question about an algebra problem by speaking into their microphone. The STEMMY platform converts speech to text, detects the user's emotional state from their facial expression, and provides an encouraging and simplified explanation.


👤 Learner
STEMMY Platform
Get Math Help via Speech
Activate Voice Input
Detect Emotion from Face
Prompt to Slow Down
Use Neutral Tone No Face Detected
🎤 Speech-to-Text Service Whisper
📷 Emotion Detection System YOLOv11
Convert Speech to Text
Classify Topic
Handle Unclear Speech
🏷️ Topic Classification Module
Retrieve Relevant Documents
Generate Explanation
🧠 LLM with RAG System Lama 3.23b
Adjust Tone Based on Emotion
Display Response
📝 Paraphrasing Module