كلية الحاسبات والذكاء الاصطناعي
FACULTY OF COMPUTERS & ARTIFICIAL INTELLIGENCE
BENHA UNIVERSITY

Emotion Detection with Timeline Analysis

A senior project submitted in partial fulfillment of the requirements for the degree of Bachelor of Computers and Artificial Intelligence.

"Computer Science" Program

Project Team

1. Ahmed Mahmoud Saeed Abdelsamie Ahmed
2. Ahmed Mohamed Fawzy Mohamed
3. Ahmed Medhat AbdelMoaz Ali
4. Ahmed Mohamed Abdallah Suleiman
5. Adam Saber Abdelaziz Moselhy Soumaa
6. Anas Alaa El-Husseiny Abdelaziz El-Gannayni
7. Fathy Ahmed Fathy El-Sayed

Under Supervision of

Dr. Mai Kamal

Benha, 2025

Abstract

Emotions are central to human communication and decision-making, influencing behavior, learning, and interaction. As technology increasingly mediates human activity, the ability of machines to detect and understand emotional states has become an essential component of modern artificial intelligence. Traditional emotion detection systems, however, often produce only a single static label—such as happy, sad, or angry—based on a brief input. They fail to capture the temporal evolution of emotions or provide meaningful insights into how an individual's emotional state changes throughout an event, conversation, or session. The **Emotion Detection with Timeline Analysis** project addresses this limitation by introducing an intelligent, multimodal system that not only identifies emotions from text, audio, image, and video but also visualizes how those emotions fluctuate over time.

The system combines advanced AI models and machine learning libraries such as **DeepFace**, **HuggingFace Transformers**, **OpenCV**, and **librosa** to process diverse input types. Each modality is analyzed independently by specialized Python microservices developed with **Flask** or **FastAPI**, and results are integrated through a central **.NET Core** backend. The frontend, built using **Angular**, provides an interactive dashboard where users can upload files, view emotion timelines, compare results across sessions, and receive graphical summaries of emotional patterns. Emotion data, including detailed frame-by-frame or segment-based results, are stored in a relational database to allow longitudinal tracking. The system also incorporates an automated alert mechanism that notifies users of recurring negative emotion patterns, encouraging emotional awareness and mental well-being.

Designed with scalability and modularity in mind, the platform supports future integration of real-time processing, mobile applications, and wearable devices. The **Emotion Detection with Timeline Analysis** system has broad applicability in fields such as education, psychology, customer experience, and human-computer interaction. By merging artificial intelligence with data visualization, it transforms emotion recognition from a static prediction task into a continuous, interpretable process that helps individuals and organizations understand emotional behavior more deeply and make informed, empathetic decisions.

Contents

Chapter 1: Introduction & Background

1.1. Introduction

Human emotions play a central role in communication, learning, decision-making, and overall well-being. In recent years, the rapid development of **artificial intelligence (AI)** and **machine learning (ML)** has made it possible to analyze, interpret, and predict human emotions from various digital inputs such as text, speech, facial expressions, and body gestures. The ability to accurately detect and understand emotions has opened new opportunities in numerous domains, including education, mental health monitoring, customer service, human-computer interaction, and social media analytics.

1.1. Motivation and Current Limitations

Despite the progress achieved by existing emotion-recognition technologies, most current systems are still limited to providing a single, static emotional output based on a specific moment of data. They lack the capacity to capture how a person's emotions **evolve over time**, which is a crucial aspect of understanding human affective behavior. Traditional systems offer only a "snapshot" of emotion, failing to provide the dynamic context necessary for deep analysis.

1.2. Project Goal and Core Innovation

The **Emotion Detection with Timeline Analysis** project aims to bridge this critical gap. The primary goal is to develop an intelligent web and mobile system capable of detecting emotions from **multiple modalities**—text, audio, images, and video—and visualizing how these emotions change throughout an interaction or over the duration of a recorded session. Rather than providing a one-time classification (e.g., "happy" or "sad"), the system's core innovation is the presentation of a **dynamic timeline of emotional states**. This allows for a much richer analysis, such as tracking how a speaker's mood fluctuates during a conversation or how a student's engagement changes during an online lecture. This continuous, temporal approach provides more meaningful understanding than traditional one-shot models.

1.3. System Architecture Overview

The system combines several interdisciplinary components into a modern, modular design:

- **AI/ML Models:** Utilizes computer vision (for facial analysis), natural language processing (for text and transcription analysis), and audio signal processing (for vocal tone) models built using frameworks like TensorFlow and PyTorch.
- **Backend Orchestration:** Developed with **ASP.NET Core**, the backend manages

authentication, media uploads, and communication with the AI modules via RESTful APIs. It is responsible for the crucial task of **timeline aggregation** and generating predictive **Active Alerts** for patterns like "High Stress" or "Mood Shift".
- **Frontend Interface:** Implemented using **Angular**, the web application provides an intuitive user experience with a Dashboard, Session History, and the core **Session Analysis** page, featuring the interactive emotion timeline visualization and actionable recommendations.

1.4. Applications and Significance

The growing significance of this project lies in extending the utility of emotion detection:

- **Healthcare:** Providing early indicators of stress, depression, or anxiety by identifying long-term emotional patterns.
- **Education:** Assisting teachers in understanding students' engagement and motivation levels during e-learning sessions.
- **Business:** Enabling organizations to analyze customer feedback videos or voice calls to assess satisfaction and improve service quality.

By adding the concept of **timeline-based emotional analysis**, the system not only identifies the dominant emotion but also monitors its transitions, allowing for **proactive interventions** and personalized support. The project represents both a technological innovation and a significant contribution to human-centered computing.

1.2. Problem Definition

In the modern digital world, interactions between humans and technology have become increasingly multimodal, incorporating text, speech, facial expressions, and gestures. However, despite the rapid advancement of artificial intelligence, most existing emotion detection systems remain limited in their ability to capture the true depth and continuity of human emotions. The problem is defined by limitations across analytical depth, data integration, temporal tracking, and user experience.

2.1. The Static Output Problem (One-Shot Classification)

Current models typically analyze a single data point—such as an image, a short audio clip, or a short text—and output one static emotion label such as "happy," "angry," or "sad". While this approach can identify a general emotional state, it fails to reflect the **dynamic nature of human feelings**, which fluctuate continuously in response to context, conversation flow, and environmental factors. This one-dimensional output provides only a **snapshot rather than a story**, preventing meaningful insight into how emotions evolve over time.

2.2. Lack of Multimodal Integration

Existing emotion recognition systems tend to focus on a **single modality of input**, such as facial expression analysis or sentiment analysis from text, without integrating multiple sources of emotional cues. This fragmentation leads to incomplete or inaccurate emotional interpretation.

- **Example of Ambiguity:** A single-modality model would misinterpret a mixed emotional signal, such as a person smiling (positive facial expression) while speaking in a frustrated tone (negative vocal emotion).

Consequently, current systems fail to deliver a holistic and reliable understanding of a user's true emotional state.

2.3. Absence of Timeline-Based Analysis

Another major limitation of conventional emotion detection systems lies in their inability to track and visualize changes in emotion throughout a temporal sequence. Emotions are inherently time-dependent phenomena that evolve gradually. When analyzing a long video, a conversation, or a speech, there may be several transitions between emotional states that are highly relevant for interpretation.

- **Loss of Valuable Variation:** Existing systems that produce only one emotion label for an entire recording obscure these valuable variations.

The absence of a timeline-based analysis prevents researchers, educators, and psychologists from observing emotional progressions and identifying key behavioral patterns, such as stress build-up, emotional fatigue, or mood improvement.

2.4. User Experience and Tracking Limitations

From a user experience standpoint, current systems also lack interactive visualization and long-term tracking features.

- **Lack of Historical Context:** Users cannot easily review their past analyses, compare emotions across multiple sessions, or detect recurring emotional trends.

The absence of personalized emotional history limits the system's usefulness for continuous self-reflection, mental health monitoring, or performance assessment.

- **Passive Technology:** The lack of alert mechanisms means that users are not notified when negative emotional patterns persist over time.

Without such features, emotion recognition remains a passive technology, offering raw classification rather than actionable emotional insight.

2.5. Technical Scalability and Interoperability Gap

There is also a technical gap in how current emotion recognition solutions are implemented.

- **Non-Production Ready:** Many research prototypes are confined to laboratory datasets and are not integrated into practical, user-friendly platforms. They often lack a scalable backend architecture that can handle large volumes of real user data, support multiple input types, and communicate effectively with AI models.
- **Monolithic Structure:** Furthermore, these systems seldom employ modular architectures that separate the frontend, backend, and AI components, which hinders maintainability and limits future extension.

The challenge, therefore, lies not only in improving emotional accuracy but also in designing a robust and scalable system that transforms complex AI outputs into an accessible, interpretable experience for everyday users.

1.3. Proposed Solution

To overcome the limitations identified in the problem definition, the proposed project—**Emotion Detection with Timeline Analysis**—introduces an intelligent, multimodal, and time-aware emotion recognition system designed to detect, analyze, and visualize human emotions from various input sources. The solution integrates artificial intelligence models for text, voice, image, and video analysis into a unified architecture that emphasizes temporal emotion tracking and user accessibility through a modern web and mobile application interface.

The core idea behind the proposed solution is to move beyond static, one-time emotion classification and instead provide users with a **timeline-based emotional analysis** that captures the natural flow of emotions over time. This approach recognizes that emotions are not isolated events but continuous processes influenced by communication, environment, and context. By representing emotions as sequences that evolve throughout a conversation, video, or audio clip, the system can provide more meaningful insights into the emotional behavior of individuals.

3.1. System Architecture and Layering

The proposed system consists of three main, separated layers to ensure scalability and maintainability:

3.1.1. Frontend Interface (User Layer)

Built using modern web technologies such as **Angular** or **React**, the frontend serves as the user interaction point. It provides pages for uploading inputs (video, image, audio, or text), viewing results, analyzing timelines, and managing personal emotion history. The design prioritizes simplicity and clarity, allowing users to easily interpret complex emotional data through **interactive charts**, visual summaries, and **color-coded emotion timelines**.

3.1.2. Backend and Integration Layer (Processing Layer)

Implemented using **ASP.NET Core**, this layer acts as the central communication hub between the user interface, database, and AI microservices. It handles data flow, user authentication, and system logic. The backend also stores analysis records, computes emotional summaries, triggers **alerts** when negative patterns are detected, and provides APIs to the frontend for visual representation.

3.1.3. AI Microservices (Intelligence Layer)

Developed in **Python** using frameworks such as **TensorFlow**, **PyTorch**, and libraries like DeepFace, librosa, and HuggingFace Transformers, these microservices perform the actual emotion detection tasks. Each modality (text, image, audio, video) is processed separately using specialized pre-trained models, and the results are standardized into a common output format.

3.2. Functional Workflow

When a user uploads an input—such as a video—the system follows a series of processing steps to generate the timeline-based emotional analysis:

1. The uploaded file is first stored securely in the server or cloud storage.
2. The backend registers the analysis in the database and forwards the file to the corresponding AI microservice.
3. The video is processed **frame by frame** using computer vision models (e.g., DeepFace or CNN-based models) to detect facial expressions and predict emotion probabilities at different time intervals.
4. If the input includes audio, the speech component is analyzed separately using audio emotion recognition techniques based on **spectral features extracted** through the librosa library.
5. For textual input, natural language processing models from HuggingFace are used to classify emotion categories such as joy, sadness, anger, fear, or surprise.
6. The backend **aggregates the results** from each modality, **aligns them temporally**, and constructs a unified emotion timeline showing how emotions shift over time.
7. The final results are stored in relational tables for detailed reporting and visualization.
8. The system computes emotion ratios (positive, negative, neutral), identifies the dominant emotion, and checks for recurring negative trends to generate alerts.
9. The frontend visualizes these results through interactive charts, such as line graphs, heatmaps, or emotion distribution pie charts, allowing users to explore how their emotions evolved across different segments of the recording.

3.3. Key Features and Innovations

The proposed solution introduces several innovative features that distinguish it from

conventional emotion detection systems:

- **Multimodal Emotion Recognition:** The system supports analysis of multiple input types—text, audio, image, and video—ensuring a more complete and accurate understanding of human emotion.
- **Timeline-Based Analysis:** Instead of a single output, the system visualizes how emotions vary across time, capturing transitions between emotional states.
- **User Emotion History:** Each analysis result is stored for every user, enabling long-term tracking of emotional changes across different sessions or interactions.
- **Alert Mechanism:** The system automatically generates alerts when persistent negative emotions or sudden drops in positivity are detected, promoting awareness and early intervention.
- **Interactive Visualization:** The use of graphs, heatmaps, and emotion timelines helps users interpret emotional trends intuitively rather than through raw data.
- **Scalable Architecture:** The separation of backend logic, AI services, and frontend interface allows easy expansion and maintenance. New models or features can be integrated without disrupting the overall system.
- **Cloud-Ready Infrastructure:** The modular design allows deployment on cloud platforms such as Azure or AWS, ensuring scalability and reliable performance for multiple users simultaneously.

3.4. Expected Benefits

The proposed solution aims to provide both technical and social benefits.

- **Technical:** It advances emotion recognition systems by introducing a structured timeline model and a unified platform for multimodal analysis. It provides researchers and developers with a practical framework to study emotional progression, dataset patterns, and cross-modal correlations.
- **Social:** The system serves as a useful tool for educators, psychologists, and organizations by offering insights into human emotional behavior. For example, educators can analyze student engagement during lessons, mental health professionals can observe mood fluctuations over time, and companies can assess customer satisfaction from recorded interactions.

In conclusion, the proposed **Emotion Detection with Timeline Analysis** system transforms traditional emotion recognition into a dynamic and interactive process that reflects the continuous and multidimensional nature of human emotion. By integrating AI-powered multimodal analysis, time-based visualization, historical tracking, and alert mechanisms, the system provides a robust, scalable, and user-centered platform for emotion analytics. This solution not only addresses the limitations of current systems but also lays the foundation for future research and practical applications in affective computing, human-computer interaction, and emotional well-being technologies.

1.4. Literature Review

4.1. Introduction

Emotion detection technologies have evolved significantly in recent years, driven by advancements in artificial intelligence, deep learning, and data analytics. Many commercial systems now offer facial expression recognition, voice-based emotion classification, and sentiment analysis from text. These tools have been adopted in industries such as customer service, marketing, education, and human–computer interaction. Despite their progress, existing systems remain limited in several critical areas: most focus on a single input modality, provide only static emotion outputs, and lack the ability to visualize how emotions change over time. Moreover, many are enterprise-oriented, inaccessible to general users, and do not offer features such as personal emotion history or alerts for negative emotional trends. This review examines the most widely used emotion detection systems across facial, speech, text, and multimodal categories, highlighting their capabilities and key limitations, and identifying the gaps that motivate the development of the proposed Emotion Detection with Timeline Analysis system.

4.2. Existing Facial Emotion Detection Systems

1- Affectiva: Affectiva is one of the most established facial emotion recognition platforms, used in advertising research, automotive applications, and audience analytics. It analyzes facial muscle movements in real time to classify emotions such as joy, anger, surprise, and confusion.

Limitations:

- Provides only moment-by-moment facial classifications with no timeline visualization for user-uploaded videos.
- Does not support multimodal processing (audio + text + video).
- No user history, trends, or alerts; results are session-based and not personalized.
- Designed primarily for enterprise use, not for public upload-and-analyze platforms.

2- Microsoft Azure Face API: Azure Face API performs face detection, emotion recognition, and facial attribute analysis. It is widely used in enterprise applications for security, retail analytics, and user experience enhancement.

Limitations:

- Offers only static emotional outputs per image or frame; no continuous emotional progression.
- Does not provide timeline graphs or emotional evolution tools.
- Single-modality: no audio, text, or multimodal analysis.
- No longitudinal tracking for individual users.

3- Google Cloud Vision AI – Face Detection: Google's Vision AI includes emotion likelihood prediction for faces in images, identifying joy, sorrow, surprise, and anger.

Limitations:

- Image-only recognition without timeline functionality.
- No session-based or historical emotional monitoring.
- No integration with other modalities or alerting mechanisms.

4.3. Existing Speech Emotion Recognition Systems

1- Cogito AI: Cogito analyzes voice features such as pitch, tone, and speaking rate to detect stress, frustration, and empathy. It is used primarily in call centers to support agents and supervisors.

Limitations:

- Domain-specific and designed for live call monitoring, not general audio uploads.
- No multimodal support and no timeline export for users.
- Not accessible as a public consumer tool.

2- CallMiner Eureka: CallMiner focuses on business call analytics, providing emotional scoring, sentiment analysis, and behavioral indicators from audio conversations.

Limitations:

- Enterprise platform restricted to large organizations.
- Does not allow user-uploaded recordings for personal emotional insight.
- Lacks multimodal integration and does not maintain emotional history outside call logs.

4.4. Existing Text Emotion and Sentiment Analysis Platforms

1- IBM Watson Tone Analyzer: Watson analyzes text to detect emotional tones such as joy, anger, sadness, and analytical writing style.

Limitations:

- Produces static emotional labels without tracking progression across long text or multi-message conversations.
- Single-modality, no timeline view, and no user-specific history.
- Not designed for multimodal alignment with facial or speech analysis.

2- Google Cloud Natural Language API: Google's NLP service performs sentiment scoring and classification over text.

Limitations:

- Limited to polarity and simple emotional categories.
- Cannot show emotional changes across paragraphs or time-based text sequences.
- No multimodal or longitudinal emotional analysis.

4.5. Existing Multimodal Emotion Recognition Systems

1- iMotions: iMotions is a powerful research-grade platform combining face analysis, voice emotion detection, eye tracking, and physiological sensors. It is widely used for academic studies and commercial research.

Limitations:

- Extremely expensive and not available for general consumers.
- Requires hardware devices for multimodal input.
- Not designed for user-uploaded content or public use.
- Timeline features exist but are restricted to controlled experiments and not personal emotion history.

2- RealEyes: RealEyes analyzes viewer reactions by detecting facial emotions while watching videos, commonly used in marketing and media testing.

Limitations:

- Focused strictly on analyzing audience reactions to advertisements.
- Provides timeline emotion graphs only for their video testing environment, not for general uploads.
- No multimodal support, no personal history, no alerts.

4.6. Existing Timeline-Based and Longitudinal Emotion Systems

1- RealEyes Timeline Analytics: RealEyes provides timeline graphs showing moment-to-moment emotional engagement during ad playback.

Limitations:

- Limited to marketing analysis; no user-uploaded videos.
- No personal emotional tracking over time.
- Does not support multimodal processing.

2- Cogito Voice Trends: Cogito shows emotional trend indicators during live customer calls (e.g., stress rising, empathy dropping).

Limitations:

- Only for voice calls; no video, no text, no images.
- No history tracking for general users.
- No timeline export or visualization for personal development.

3- Affectiva Automotive: Used in cars to detect driver fatigue and distraction, providing continuous monitoring.

Limitations:

- Designed for automotive integration, not general access.
- Does not support user uploads.
- No cross-modality analysis, no emotional history, no alerts for everyday users.

4.7. Timeline and Longitudinal Emotion Analysis Gaps

- **Lack of Continuous Emotion Tracking:** Most existing emotion detection tools provide isolated emotion outputs without showing how emotions evolve throughout a video, audio segment, or conversation. This absence of continuous tracking hides important emotional fluctuations, transitions, and intensity changes that are essential for meaningful interpretation.

- **No Unified Timeline Across Modalities:** Current systems do not combine different modalities (facial expressions, speech tone, and textual content) into a single synchronized timeline. Each modality is analyzed separately, resulting in fragmented emotion insights rather than an integrated emotional narrative.
- **Absence of User-Uploaded Timeline Analysis:** Timeline-based applications such as RealEyes or Cogito operate only within their controlled environments and datasets. They do not allow general users to upload personal videos, audio recordings, or text sessions and receive a personalized timeline analysis, significantly limiting accessibility.
- **Missing Long-Term Emotional History:** Existing systems treat every analysis as a standalone session. None store emotional results across multiple uploads, preventing users from observing long-term emotional patterns, mood stability, or recurring negative states over weeks or months.
- **No Alert Mechanisms for Repeated Patterns:** Timeline-based systems do not incorporate detection of negative emotional trends, such as prolonged sadness, rising stress, or sudden shifts in emotional stability. Without alerts or warnings, users cannot benefit from early emotional insights or preventative well-being measures.
- **Domain-Specific or Enterprise-Only Tools:** Current timeline-capable platforms are limited to specific industries (marketing analytics, call centers, automotive safety). They are unavailable to the general public and do not serve personal emotion monitoring or mental well-being use cases.
- **Lack of User-Friendly Visualization Tools:** Most existing systems that offer any form of timeline visualization do so through complex research dashboards or enterprise interfaces. They lack intuitive, accessible, web-based tools suitable for everyday users, students, or non-technical individuals.

4.8. Combined Limitations of Existing Systems

Across all categories—facial, speech, text, multimodal, and timeline-based—several recurring limitations exist:

1. **Lack of multimodal integration:** Most systems analyze only one modality (image, audio, or text), not a unified emotional profile.
2. **Absence of timeline-based emotion visualization:** Few platforms display emotional changes over time, and those that do are not accessible to the general public.
3. **No personal emotional history or long-term tracking:** Existing tools do not store results across multiple sessions to reveal emotional trends.
4. **No alert mechanisms:** No system warns users about recurring negative emotions or sudden emotional drops.
5. **Enterprise-focused rather than user-friendly:** Most powerful systems are designed for business or research institutions, not everyday users.
6. **No unified platform for all input types:** No current system allows users to upload text,

audio, image, and video in one place and receive a combined analysis.

These gaps highlight a major need for a practical, accessible, multimodal system capable of timeline visualization and long-term emotional monitoring.

4.9. Summary

Existing emotion detection technologies demonstrate impressive capabilities within isolated domains such as facial recognition, speech analysis, or textual sentiment extraction. However, they remain limited in accessibility, multimodal integration, timeline visualization, and longitudinal emotional tracking. None of the reviewed systems offer a unified platform where general users can upload multiple types of media and receive detailed emotion timelines, historical tracking, and alerts for negative emotional trends. The proposed Emotion Detection with Timeline Analysis system fills this gap by integrating multimodal emotion detection with continuous timeline visualization and personal emotional history, offering a comprehensive, user-centered platform not available in existing solutions.

1.5. Project Objective

The primary objective of the **Emotion Detection with Timeline Analysis** project is to develop an intelligent, web-based multimodal emotion recognition system capable of analyzing text, audio, image, and video data to understand and visualize emotional changes over time. Unlike conventional systems that generate static emotion predictions, this system focuses on capturing the dynamic progression of emotions, providing users with meaningful insights into emotional fluctuations and patterns.

Detailed Objectives

- **Comprehensive Emotion Recognition:** Detect emotions from multiple input sources (text, speech, image, and video) using specialized pre-trained deep learning models and libraries such as **DeepFace**, **HuggingFace**, and **librosa**.
- **Temporal Emotion Tracking:** Implement a **timeline-based emotion analysis method** that shows how emotions change throughout a session, conversation, or video, rather than producing a single, fixed output.
- **Emotion Visualization:** Provide a graphical dashboard with interactive charts, line graphs, and emotion heatmaps to represent emotional progression and variations over time.
- **User Dashboard and History:** Design a user-friendly web interface that allows users to upload or record data, view detailed emotion reports, and maintain a **personal history of all analyses** for future reference and comparison.
- **Automated Alerts:** Introduce an intelligent alert system that detects recurring negative emotional patterns and generates notifications or warnings, helping users identify potential stress or mood deterioration.

- **AI Model Integration:** Integrate pre-trained AI models using frameworks such as **DeepFace** for facial expression recognition, **HuggingFace Transformers** for text and speech emotion detection, **OpenCV** for video frame extraction, and **librosa** for audio feature analysis.
- **Scalable and Modular Architecture:** Develop a modular, service-oriented system architecture that supports scalability, easy maintenance, and future expansion. Potential future extensions include real-time emotion tracking and integration with mobile or wearable devices for continuous monitoring.

In summary, the project aims to provide a comprehensive emotion analysis platform that bridges artificial intelligence, human emotion understanding, and data visualization. It not only enhances emotion detection accuracy but also promotes emotional awareness and digital well-being by revealing patterns hidden across time and context.

1.6. Scope of the Project

The **Emotion Detection with Timeline Analysis** project encompasses the design, development, and implementation of a web-based application (with planned mobile extension) that performs multimodal emotion detection and time-based emotional analysis using artificial intelligence and machine learning technologies. The project emphasizes usability, accuracy, and visualization of emotional changes over time.

In-Scope Features

- **Frontend (Angular):** The web interface is developed using **Angular** and styled with **TailwindCSS** or **Angular Material**. It allows users to upload videos, images, audio, or text and visualize emotion timelines using dynamic, interactive charts and graphs.
- **Backend (.NET Core API):** The backend, built with **ASP.NET Core Web API**, manages user authentication, session handling, and data communication between the frontend, AI microservices, and the database. It ensures reliable data flow and optimized performance.
- **AI Microservices (Python):** The emotion recognition models are implemented as independent **Python microservices** using frameworks such as **Flask** or **FastAPI**. Each service handles a specific input modality—text, audio, image, or video—and returns structured JSON emotion results.
- **Database (SQL Server / PostgreSQL):** The database stores user data, analysis records, emotion timelines, summary statistics, and alerts. It also supports historical tracking, allowing users to view and compare past emotion analyses.
- **Visualization and Analytics:** The system presents emotional variations and trends over time through graphical visualizations, reports, and timeline charts. This feature helps users understand emotional progression and identify behavioral patterns.
- **User Roles and Accessibility:** The system is designed to accommodate different user

types such as general users, researchers, educators, and professionals interested in emotion monitoring and analysis. It emphasizes accessibility, clarity, and ease of interpretation.

- **Mobile Extension (Future Phase):** As part of the future system expansion, a **Flutter-based mobile application** will be developed to complement the web system. The mobile app will enable users to:
  - Record or upload text, audio, image, or video data directly from smartphones.
  - View emotion detection results and emotion timelines through a mobile-optimized interface.
  - Receive push notifications or alerts when consistent negative emotion patterns are detected.

The mobile version will share the same backend and AI microservices as the web platform, ensuring full synchronization and data consistency across all devices.

Out of Scope (Phase 1)

- Real-time emotion detection from live video streams or webcam feeds.
- Integration with wearable or physiological sensors for biometric emotion recognition.
- Multi-language model training and cultural emotion adaptation.
- Cloud-based large-scale deployment or optimization for enterprise usage (reserved for later phases).

Summary

This project delivers a complete, scalable ecosystem for multimodal emotion detection and timeline-based emotion analysis. The combination of web and planned mobile interfaces ensures accessibility and convenience for users. By incorporating AI-driven analytics and visualization, the system provides deep emotional insights, allowing users to observe how their emotions evolve and receive alerts for potential negative patterns. This version focuses on achieving accurate emotion detection, timeline visualization, and modular architecture, while laying the groundwork for future real-time and mobile enhancements.

1.7. Scope Exclusions and Constraints

Although the Emotion Detection with Timeline Analysis project aims to develop a robust and intelligent multimodal emotion recognition system, certain features and functionalities remain beyond the scope of the current phase due to academic, technical, and resource limitations. The following exclusions and constraints define the project boundaries:

Scope Exclusions

- **Real-time continuous tracking:** Live streaming or webcam-based real-time emotion detection will not be implemented in this version. Instead, the system focuses on

analyzing uploaded videos, images, text, or audio samples.

- **Model training from scratch:** The project will not involve training models from raw datasets. Instead, pre-trained models such as DeepFace and HuggingFace Transformers will be utilized to reduce computational complexity and allow more focus on system integration and visualization.
- **Physiological emotion detection:** Emotion recognition based on physiological signals such as heart rate, EEG, or galvanic skin response is excluded. The project focuses solely on visual, audio, and textual modalities.
- **Mobile or wearable applications:** Mobile versions or wearable integrations will be considered in future work. The current scope includes only the web-based platform.
- **Psychological diagnosis or medical evaluation:** The system is not intended for clinical or psychological diagnosis. It provides analytical visualizations of emotional states for awareness and research purposes only.

Constraints

- **Time Constraint:** The project is developed within a limited academic schedule, requiring efficient task management and milestone-based development.
- **Hardware Constraint:** Development and testing will be conducted using standard hardware without access to high-performance GPUs. This limitation may restrict large-scale model processing and reduce response time for large files.
- **Data Constraint:** The project will rely on publicly available emotion datasets for testing and validation, which may not fully represent all cultural and demographic variations.
- **Privacy Constraint:** Raw user data such as uploaded images, audio, video, or text will not be permanently stored. However, processed emotion results and summaries will be maintained to preserve user analysis history while ensuring privacy.
- **Environmental Constraints:** External factors such as lighting variations, background noise, or low-quality media may slightly reduce detection accuracy. In addition, cultural differences in emotional expression may influence model predictions.

1.8. Project Methodology

The Emotion Detection with Timeline Analysis system will be developed following the **Agile Software Development Life Cycle (SDLC)** model. This methodology promotes iterative development, continuous feedback, and adaptability, ensuring that the project evolves efficiently through incremental improvements and testing.

1. Planning Phase

During this phase, the overall project vision, goals, and success criteria were defined. The team identified the need for a system capable of detecting emotions from multiple modalities (text, voice, image, and video) and visualizing how these emotions change over time. Key resources, technologies, and frameworks were selected, including **Angular** for the frontend,

.NET Core for the backend, and **Python-based AI microservices**. The user interaction flow and system layout were also outlined to ensure usability and intuitive design.

2. Analysis Phase

In this phase, both functional and non-functional requirements were identified. Functional requirements included multimodal emotion detection, timeline generation, visualization, and alert mechanisms. Non-functional requirements addressed scalability, performance, privacy, and usability. Existing emotion analysis tools, such as Affectiva and Microsoft Azure Face API, were reviewed to identify their limitations. This analysis guided the decision to combine multimodal emotion recognition with timeline-based visualization for improved interpretability.

3. Design Phase

This stage focused on translating requirements into a structured system design and architecture.

- **Frontend Design:** Developed using Angular and TailwindCSS to provide a responsive, visually appealing, and user-friendly interface for uploading data and viewing results.
- **Backend Design:** Built using .NET Core Web API to manage user authentication, data flow, and communication between the AI microservices and the frontend.
- **AI Microservices:** Implemented using Python with Flask or FastAPI to perform emotion recognition tasks. **DeepFace** handles facial emotion analysis, **HuggingFace Transformers** support text and speech emotion detection, while **OpenCV** and **librosa** are used for segmenting and analyzing video and audio data for frame-by-frame evaluation.
- **Database Design:** Structured to store processed emotion results, summaries, and user history data securely, ensuring that sensitive raw input is handled with privacy safeguards.

4. Implementation Phase

This phase involved coding, component integration, and iterative testing using Agile sprints. Each sprint delivered a functional module, such as emotion detection, timeline generation, or dashboard visualization. The AI microservices were connected to the backend through **RESTful APIs** for efficient data exchange. The backend processed and aggregated AI outputs, transforming them into structured results for the frontend. The frontend displayed interactive charts and graphs representing emotional progression over time, using data retrieved from the backend.

5. Testing Phase

A comprehensive testing strategy was adopted to ensure functionality, accuracy, and system

reliability. This included **Unit Testing**, **Integration Testing**, **System Testing**, and **User Acceptance Testing (UAT)**. Testing also verified that the detected emotions aligned with realistic human interpretation and that the system met both functional and non-functional requirements.

Deployment and Maintenance

Following successful testing, the system will be deployed as a web-based application accessible through standard browsers. Deployment involves configuring the backend, AI microservices, and frontend for seamless interaction. Future maintenance activities will include bug resolution, updating pre-trained models for improved accuracy, and adding advanced features such as real-time emotion detection and wearable device integration to enhance system capabilities.

Chapter 2: Project Management

2.1. Project Organization

The Emotion Detection with Timeline Analysis project is structured into three major components — Frontend, Backend, and AI Services — each focusing on specific responsibilities to ensure smooth development, clear communication, and effective integration.

Team Structure

- **Project Manager:** Oversees project planning, task assignment, and overall coordination among team members. Ensures that milestones and deadlines are achieved according to the project timeline.
- **Frontend Developers (Angular & Flutter):** Responsible for designing and implementing the user interfaces for both web and mobile applications.
    - The Angular team focuses on the web dashboard and emotion visualization features.
    - The Flutter team develops a cross-platform mobile application that mirrors the web functionality.
- **Backend Developers (.NET Core):** Develop RESTful APIs, handle data exchange between frontend and backend, manage user sessions, and ensure secure integration with AI microservices.
- **AI & Data Engineers (Python):** Design and integrate machine learning and deep learning models for emotion recognition across text, audio, image, and video. Use frameworks such as DeepFace, HuggingFace, OpenCV, and librosa.
- **Database Administrator:** Manages the SQL Server database structure, ensuring data consistency, indexing efficiency, and secure data access policies.
- **UI/UX Designer:** Creates intuitive, accessible, and user-friendly designs that align with usability principles for both web and mobile platforms.
- **Quality Assurance (QA) Engineer:** Tests system functionality, checks for bugs or performance issues, and ensures all features meet accuracy and quality standards.

Communication Workflow

- Weekly sprint reviews and team meetings to track progress.
- Collaboration tools such as Trello, Jira, or GitHub Projects for task management.
- Source control and versioning handled through Git and GitHub repositories.
- Continuous Integration (CI) and Continuous Deployment (CD) pipelines for automated testing and updates

2.2. Risk Management

Effective risk management is crucial for maintaining the quality, schedule, and reliability of the project. The team identifies, assesses, and mitigates potential risks at every development phase to ensure project stability.

Potential Risks and Mitigation Strategies

Risk Type	Description	Mitigation Strategy
Technical Risk	AI models may not reach expected accuracy or may fail to generalize across all emotion types and data inputs.	Use reliable pre-trained models, perform iterative testing, and fine-tune parameters for each modality.
Integration Risk	Communication errors or mismatches between backend APIs and AI microservices.	Establish consistent data formats, implement versioned APIs, and conduct integration testing early.
Performance Risk	Processing large video or audio files might cause slow responses or system lag.	Optimize data pipelines, use caching, and leverage asynchronous task processing.
Data Risk	Possibility of user data loss, corruption, or security breaches.	Apply encryption, backup policies, and secure authentication methods.
Schedule Risk	Delays in deliverables due to unforeseen issues or resource limitations.	Use agile methodology, maintain flexible scheduling, and monitor progress through regular reviews.
Operational Risk	Unpredicted bugs, deployment errors, or hardware/software failures.	Maintain rollback mechanisms, automated testing, and version control across all environments.

Summary

A well-defined organizational structure and strong risk management framework are key to achieving the project's objectives successfully.By clearly defining roles and anticipating potential risks, the team ensures efficient collaboration, timely delivery, and a reliable emotion detection system that performs consistently across all platforms.

2.3. PROJECT COMMUNICATION PLAN

Stakeholder	Deliverable	Frequency	owner	Preferred Way to Deliver	Notes & Attachments
Project Team	Idea of the project	One time	D. Mai Kamal	Team meeting	Include a Presentation for the idea
Project Team	Idea of the project	One time	Eng. Fatma Ebrahim	Team meeting	Explain the idea of the project to Eng. Fatma
Project Team	Completed tasks	Every Thursday at 10 P.M.	Project Team	Discord	Review and evaluate the status of the project
Project Team	Updated work for the project	Every two weeks	D. Mai Kamal	Team meeting	Explanation of the updated work
Project Team	Updated work for the project	Every Monday at 3 P.M .	Eng. Fatma Ebrahim	Team meeting	Explanation of the updated work

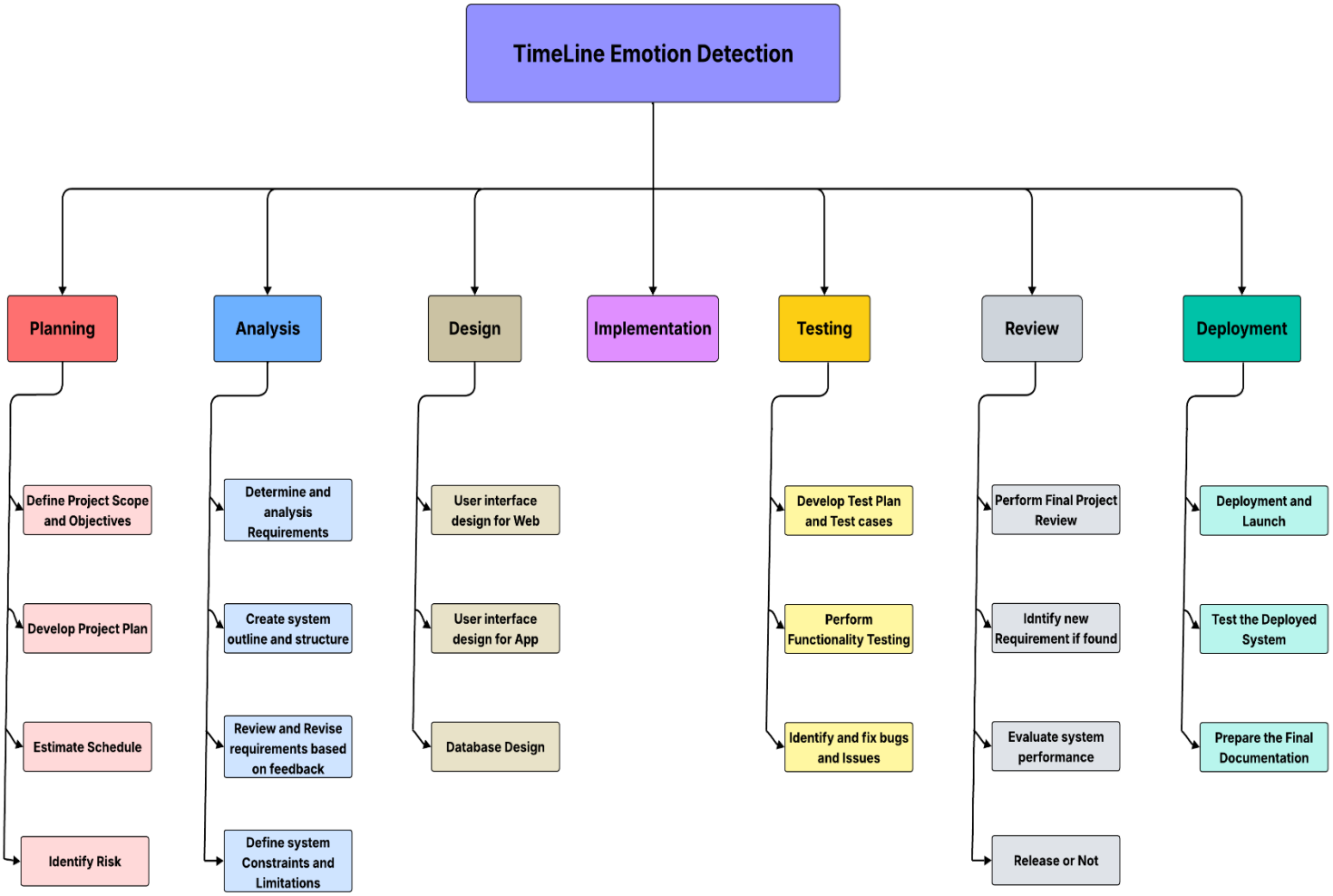
2.4. WORK BREAKDOWN STRUCTURE (WBS)

The Work Breakdown Structure (WBS) is a fundamental project management tool that serves as a visual representation of the project's hierarchical decomposition. It provides a systematic and organized breakdown of the project into manageable and understandable components. Each component represents a specific task, deliverable, or work package. WBS serves as a communication tool, ensuring that all stakeholders have a common understanding of the project's structure and components .

The Work Breakdown Structure (WBS) holds significant importance for developing Timeline Emotion Detection System :

- **Scope Definition :** The WBS plays an essential role in clearly outlining the scope of the Timeline Emotion Detection System. It ensures that all project components—such as AI emotion analysis, timeline visualization, backend services, and mobile features—are fully understood by both the team and stakeholders.
- **Facilitates Communication :** By presenting the project in a structured and visual breakdown, the WBS significantly improves communication among developers, designers, testers, and project managers. It enhances alignment by illustrating how each system component fits within the overall project framework.
- **Task Distribution :** The WBS divides the project into smaller, manageable tasks such as frontend development, machine learning model training, database setup, and UI/UX creation. This helps assign responsibilities efficiently, increases productivity, and ensures that each team member understands their specific role.
- **Dependency Identification :** Using the WBS helps identify dependency relationships between tasks—such as backend API readiness before mobile integration, or dataset preparation before AI model training. Recognizing these dependencies supports precise scheduling and a smoother workflow.
- **Progress Monitoring :** Breaking the project into standardized units allows for accurate monitoring of progress. Each module—web interface, mobile app, backend API, machine learning model—can be evaluated individually, making it easier to detect delays or issues and take corrective actions.
- **Schedule Planning :** The WBS provides a foundation for building a detailed timeline that includes milestones such as requirement completion, UI/UX approval, model training phases, integration testing, and deployment. It supports realistic scheduling and helps ensure deadlines are met efficiently.

In summary, the Work Breakdown Structure is a foundational tool for ensuring the successful development and implementation of the Timeline Emotion Detection System (Web & Mobile Application) . It clarifies scope, strengthens collaboration, enhances task management, and contributes to precise planning and execution.



2.5. TIME MANAGEMENT

2.5.1. PERT Equation

A technique that uses optimistic, pessimistic, and realistic time estimates to calculate the expected time for a particular task.

The Equation : **ET = (o + 4 * r + p) / 6**
Where :

- ET = expected time for the completion of the task.
- O = optimistic completion time for the task.
- R = realistic completion time for the task.
- P = pessimistic completion time for the task.

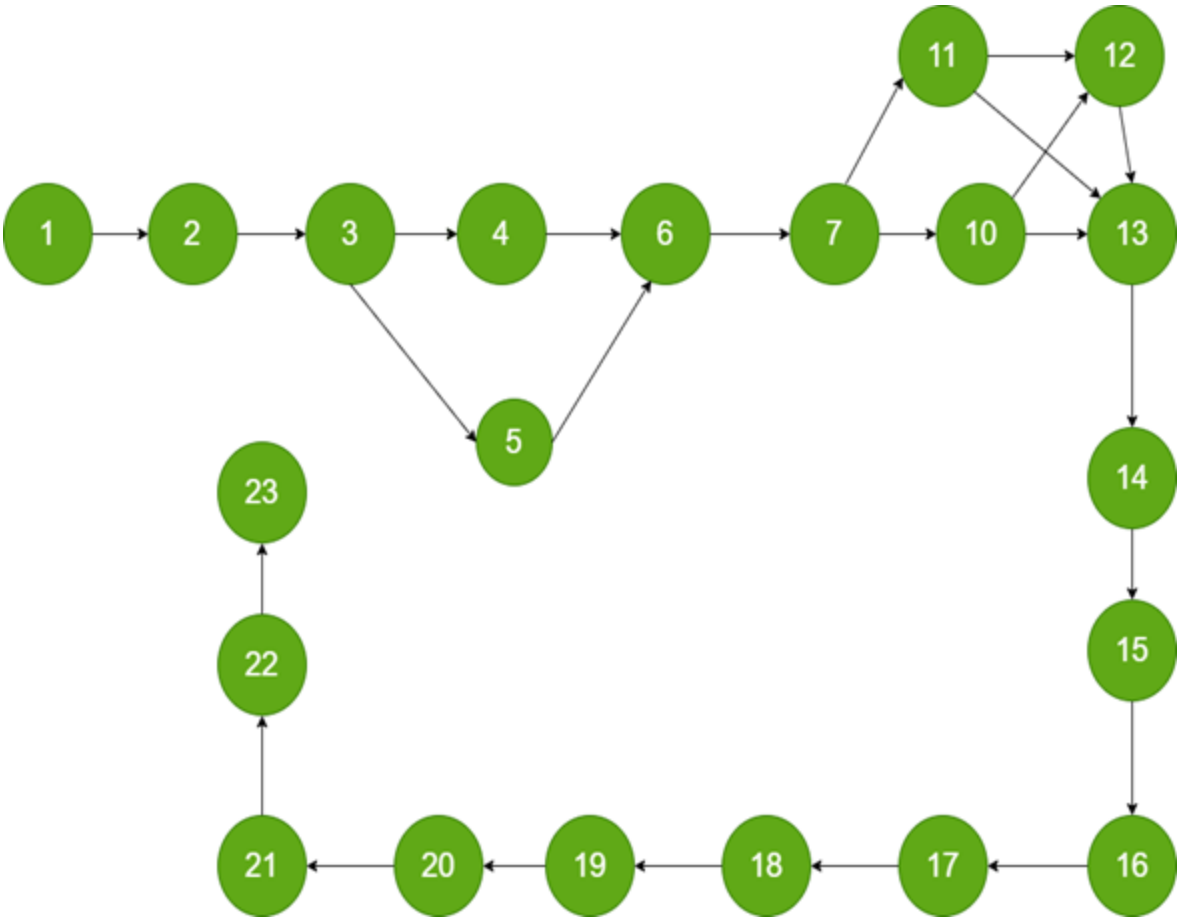
Note that: the time on the table is in Days.

Task Number	Task	o	r	p	ET
1	Define project Scope and Objectives	3	5	7	5
2	Develop Project Plan	2	4	6	4
3	Estimate Schedule	1	2	3	2
4	Identify Risk	2	4	6	4
5	Determine and analysis requirements	4	6	8	6
6	Create system outline and structure	2	3	4	3
7	Review and revise requirements based on feedback	2	3	4	3
8	Define system constraints and Limitations	1	3	5	3
9	Estimate Cost	1	2	3	2
10	User interface design for Web	5	7	9	7
11	User interface design for App	5	7	9	7
12	Database Design	4	5	6	5
13	Implementation	27	30	33	30
14	Develop Test plan and Test cases	2	4	6	4
15	Perform Functionality Testing	5	7	9	7
16	Identify and Fix bugs and Issues	7	9	11	9
17	Perform Final project Review	2	3	4	3
18	Identify new Requirement if found	3	5	7	5
19	Evaluate system performance	2	3	4	3
20	Release or Not	1	1	1	1
21	Deployment and Launch	1	3	5	3
22	Test the Deployed System	2	4	6	4
23	Prepare the Final Documentation	3	4	5	4

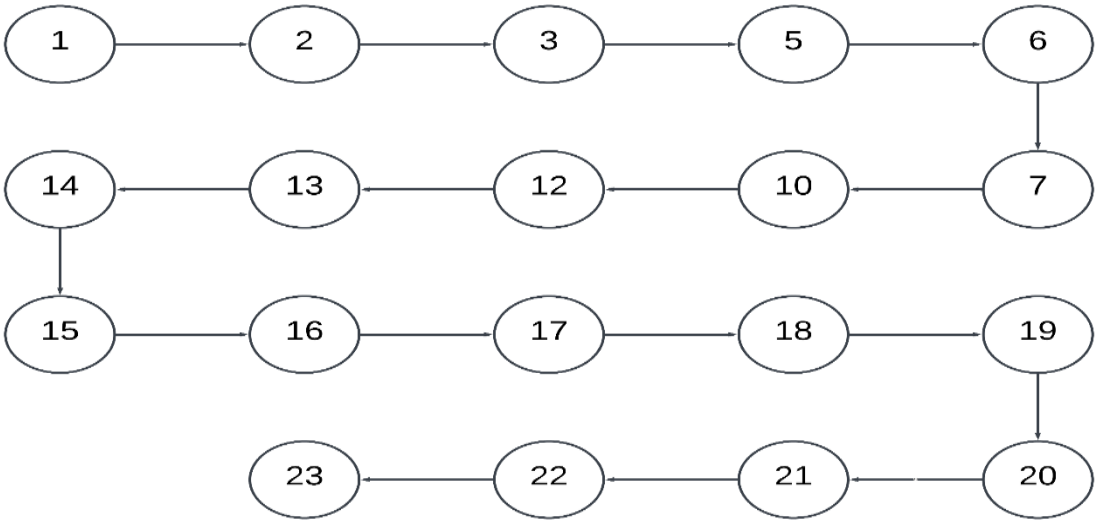
2.5.2. Network diagram

The Network Diagram is a visual representation of the project activities arranged in a order that shows how tasks are connected, the sequence in which they must be performed, and the dependencies between them. It helps project managers understand the project flow, identify critical paths, estimate project duration, and recognize possible delays.

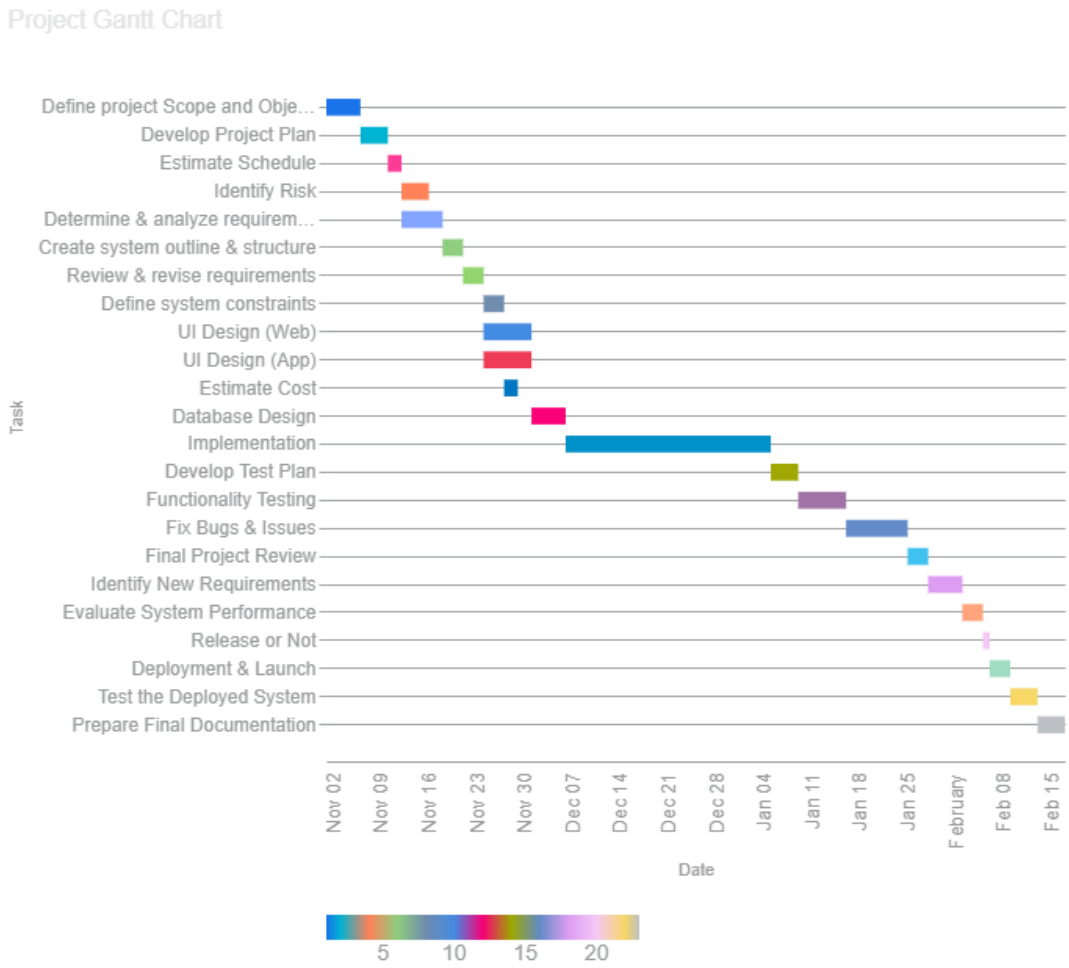
Task Number	Task	time	Depends On
1	Define project Scope and Objectives	5	---
2	Develop Project Plan	4	1
3	Estimate Schedule	2	2
4	Identify Risk	4	3
5	Determine and analysis requirements	6	3
6	Create system outline and structure	3	5,4
7	Review and revise requirements based on feedback	3	6
8	Define system constraints and Limitations	3	7
9	Estimate Cost	2	8
10	User interface design for Web	7	7
11	User interface design for App	7	7
12	Database Design	5	11+10
13	Implementation	30	10 + 11 + 12
14	Develop Test plan and Test cases	4	13
15	Perform Functionality Testing	7	14
16	Identify and Fix bugs and Issues	9	15
17	Perform Final project Review	3	16
18	Identify new Requirement if found	5	17
19	Evaluate system performance	3	18
20	Release or Not	1	19
21	Deployment and Launch	3	20
22	Test the Deployed System	4	21
23	Prepare the Final Documentation	4	22


1
2
3
4
5
6
7
10
11
12
13
14
15
16
17
18
19
20
21
22
23

2.5.3. Critical path



2.5.4. Gantt Chart



Chapter 3: System Analysis

3.1 Functional Requirements

This document outlines the comprehensive functional requirements for the Emotion Detection with Timeline Analysis system. The requirements are categorized into core domains: User Identity & Security, Multimodal AI Services, Temporal Analysis & Visualization, and Administrative Governance.

3.1.1 Domain I: User Identity, Security, and Session Management

3.1.1.1 Secure Account Onboarding and Registration

- **FR-1.1.1: Multi-Step Registration**: The system shall provide a secure registration workflow requiring a unique email address, a distinctive username, and a high-entropy password.
- **FR-1.1.2: Real-Time Input Validation**: The interface must perform live validation of email formats and password complexity (entropy score), providing immediate corrective feedback to the user.
- **FR-1.1.3: Mandatory Email Verification**: To prevent bot registration and ensure data integrity, the system shall issue an automated verification link. Full platform access is restricted until the account is verified.

3.1.1.2 Authenticated Access and Session Security

- **FR-1.2.1: Secure Login Protocols**: The system shall facilitate authenticated sessions using either a username or email paired with a secure password.
- **FR-1.2.2: Brute-Force Mitigation**: The platform must implement rate-limiting and account lockout mechanisms to defend against automated credential-stuffing and brute-force attacks.
- **FR-1.2.3: Secure Credential Recovery**: A self-service "Forgot Password" feature shall be provided, utilizing time-sensitive, authenticated email tokens for secure password resets.

3.1.1.3 Advanced Identity Verification and Privacy

- **FR-1.3.1: Risk-Based Re-Authentication**: For sensitive operations—such as accessing deep historical emotional data or responding to high-risk alerts—the system shall require secondary verification.
- **FR-1.3.2: Multi-Factor Authentication (MFA)**: The system shall support 2FA options, including One-Time Passwords (OTP) via SMS or email, to enhance account security.
- **FR-1.3.3: Granular Privacy Configuration**: Users shall have the authority to manage data retention policies, toggle timeline visibility, and configure notification preferences based on alert severity.

3.1.2 Domain II: Multimodal Emotion Detection Services

3.1.2.1 Textual Affective Analysis

- **FR-2.1.1: Bulk Text Processing**: The system shall support the input and analysis of textual content up to a limit of 10,000 characters per session.
- **FR-2.1.2: Deep Emotional Classification**: The AI engine must identify a dominant emotion, detect secondary emotional nuances, and provide a mathematical confidence percentage for each classification.
- **FR-2.1.3: Contextual Persistence**: Users shall have the option to save text analyses to their timeline, including the raw input and a personal "contextual note" for future reference.

3.1.2.2 Acoustic and Vocal Emotion Recognition

- **FR-2.2.1: Audio Ingestion**: The platform shall support the upload or direct recording of audio files (up to 5 minutes) for emotional evaluation.
- **FR-2.2.2: Temporal Vocal Mapping**: The system shall extract acoustic features to generate a time-stamped emotional chart, showing fluctuations in vocal affect over the duration of the clip.
- **FR-2.2.3: Interactive Playback Interface**: An integrated player shall allow users to listen to their audio while viewing a synchronized overlay of the emotional analysis results.

3.1.2.3 Visual Facial Expression Detection (Image)

- **FR-2.3.1: Multi-Face Detection**: The AI module shall identify and isolate one or more human faces within a single image file for individual emotional classification.
- **FR-2.3.2: Privacy Redaction**: Depending on user-defined privacy settings, the system shall provide the capability to redact or blur facial previews in the timeline view.

3.1.2.4 Dynamic Video-Based Timeline Analysis

- **FR-2.4.1: Video Format Compatibility**: The system shall support standard video containers (MP4, MOV, AVI) for frame-by-frame emotional evaluation.
- **FR-2.4.2: Temporal Heat-Mapping**: The platform shall generate emotional heat maps and summary reports that visualize the intensity and progression of emotions throughout the video duration.
- **FR-2.4.3: Segment-Specific Analysis**: Users shall be able to define custom "Start" and "End" timestamps to analyze specific segments of a longer recording.

3.1.3 Domain III: Temporal Visualization and Trend Analytics

3.1.3.1 The Emotional Timeline

- **FR-3.1.1: Centralized Chronological Ledger**: The system must maintain a unified, auditable record of all emotional analyses across text, audio, image, and video

modalities.

- **FR-3.1.2: Lifecycle Management**: Users shall have the ability to search, filter, update (notes), or soft-delete entries within their emotional history.

3.1.3.2 Longitudinal Trend Identification

- **FR-3.2.1: Statistical Aggregation**: The system shall aggregate data points over weeks or months to detect statistically significant behavioral shifts.
- **FR-3.2.2: Interactive Graphic Export**: The platform shall generate exportable charts (line graphs, scatter plots) that allow users to visualize emotional stability or volatility over custom date ranges.

3.1.3.3 Proactive Alerting and Intervention

- **FR-3.3.1: Abnormal Pattern Detection**: An intelligent algorithm shall monitor for concerning patterns, such as a consistent rise in stress indicators or prolonged periods of high-intensity negative emotions.
- **FR-3.3.2: Severity-Based Notification**: The system shall classify risks (Low, Medium, High) and issue multi-channel alerts (Email, Push, In-App) accompanied by supportive wellness resources.

3.1.4 Domain IV: System Administration and Resource Governance

3.1.4.1 User and Role-Based Administration (RBAC)

- **FR-4.1.1: Administrative Oversight**: Authorized admins shall have the power to view, modify, deactivate, or remove user accounts via a dedicated management dashboard.
- **FR-4.1.2: Immutable Activity Logging**: The system shall record all administrative actions to ensure accountability and maintain a secure audit trail.

3.1.4.2 AI Model and Alert Management

- **FR-4.2.1: Model Versioning and Deployment**: Administrators shall be able to upload, retrain, or switch between active AI model versions (e.g., swapping a facial detection model).
- **FR-4.2.2: Threshold Sensitivity Tuning**: Admins shall have the authority to adjust the mathematical thresholds that trigger "High Risk" alerts globally to minimize false positives.

3.1.4.3 Data Synchronization and Infrastructure

- **FR-4.3.1: Cross-Platform Synchronization**: The system must ensure that all timeline data and user settings are synchronized between the Web (Angular) and Mobile (Flutter) applications using a conflict-resolution strategy.
- **FR-4.3.2: Media Handling and Encryption**: The platform shall enforce strict file-size limits and provide descriptive error messages for unsupported formats. All uploaded media must be encrypted at rest and in transit.

3.2 Non-Functional Requirements (NFR)

The Non-Functional Requirements (NFR) section establishes the foundational quality framework for the Emotion Detection with Timeline Analysis platform. While functional requirements describe the discrete actions the system must perform, NFRs define the constraints, standards, and performance benchmarks that ensure the system operates with professional-grade stability, security, and efficiency. In the context of an affective computing platform, these requirements are critical as they govern the handling of sensitive biometric data and the reliability of emotional insights that could impact user well-being.

3.2.1 Domain I: Security and Data Sovereignty

Security is the primary non-functional pillar. The platform must protect highly sensitive assets, including facial data, vocal signatures, and chronological emotional reflections.

3.2.1.1 Comprehensive Asset Identification

To implement a granular security strategy, the platform identifies and classifies all digital assets into the following critical categories:

- **User Emotional Intelligence Assets**: This encompasses the "Emotional History Timeline," which contains all recorded entries across text, audio, image, and video modalities. It also includes "Sensitive Emotional Alerts"—high-risk flags, mental-health notifications, and the outputs of longitudinal trend analysis.
- **Identifiable Personal Information (PII)**: Includes profile data such as names, unique email addresses, phone numbers, and demographic data. Crucially, this also includes "Authentication Information" such as cryptographically hashed passwords, Multi-Factor Authentication (MFA) tokens, and time-sensitive One-Time Passwords (OTP).
- **Multimodal Media Resources**: Raw files uploaded for analysis (MP4, AVI, WAV, PNG, JPG). This includes "Sensitive Visual Data" where faces are isolated from video frames or static images, and acoustic signatures extracted from voice recordings.
- **Infrastructure and AI Intellectual Property**: Proprietary emotion detection models (text, audio, image, video), their training weights, performance metrics, and the backend architecture (API servers, GPU clusters, and relational databases).
- **Governance and Audit Assets**: Comprehensive "Audit Logs" (login attempts, analysis triggers, model version updates) and "Security Logs" (failed authentication patterns, suspicious file uploads, and anomaly detection results).

3.2.1.2 Threat Landscape and Exposure Analysis

The system architecture is designed to proactively mitigate specific exposure vectors identified during the analysis phase:

- **Data and Metadata Exposure**: We mitigate the risk of unprotected emotional history

and unencrypted media files. Without encryption, an attacker could reconstruct a user's psychological profile.

- **Access and Identity Exposure**: Addressing "Weak Authentication" (poor password entropy) and "Unauthorized Horizontal Access," where a malicious user might attempt to exploit API endpoints to view another user's private timeline.
- **Platform and Logic Vulnerabilities**: Mitigating software bugs in the backend or AI inference endpoints. A major focus is placed on "Insecure File Handling," preventing malicious uploads that might attempt to exploit vulnerabilities in media processors (e.g., OpenCV or librosa).
- **Human and Administrative Exposure**: Implementing strict protocols against "Social Engineering," where users might be manipulated into sharing their emotional reports, and "Admin Misuse," ensuring that any administrative access to user data requires an auditable justification.

3.2.1.3 Vulnerability Management and Control Protocols

The platform achieves a robust security posture through the following control mechanisms:

- **Vulnerability Avoidance**: The system employs regular security audits, monthly vulnerability scans on API services, and periodic penetration testing specifically targeting media upload endpoints. We adhere to OWASP standards for web and API security.
- **Active Attack Detection**: Implementation of "Impersonation Detection," where the system triggers device-based verification (OTP) if a login attempt originates from an unrecognized hardware signature.
- **Malware Neutralization**: The media pipeline includes a sanitization layer that scans for malware embedded in video/audio metadata (e.g., EXIF injection) before the file reaches the AI models.
- **Encryption and Rollback**: End-to-end encryption for data in transit and at rest ensures that even if exposure occurs, the data remains unreadable. Furthermore, the system supports "Rapid Recovery" via rollback features to restore user data to a stable state after an incident.

3.2.2 Domain II: Availability, Reliability, and Fault Tolerance

Availability ensures that the platform is a dependable companion for emotional monitoring, maintaining accessibility 24/7.

3.2.2.1 High Availability and Redundancy Strategies

- **Service Level Agreements (SLAs)**: The platform aims for 99.5% availability. This includes established benchmarks for maximum delay in timeline updates and strict time limits for the delivery of emotional alerts.
- **Redundant Architecture**: Using redundant application servers and database replicas (Primary-Secondary clusters) ensures that if a server node fails, the system continues to

function without user-perceived downtime.

- **Load Balancing and Concurrency**: High-volume media uploads—particularly AI-heavy video processing—are distributed across multiple GPU-enabled processing nodes using intelligent load balancers to prevent system saturation.

3.2.2.2 Fault Management and Reliability Achievement

Reliability is defined as the system's ability to consistently process emotions without errors or interruptions.

- **Error vs. Fault vs. Failure Analysis**:
  - **Human Error**: The system handles "Human Mistakes" (e.g., uploading a blurred image or a corrupted file) by providing clear, real-time feedback and corrective guidance rather than crashing.
  - **System Faults**: Technical malfunctions such as server outages or API timeouts are mitigated through "Automatic Failover" mechanisms and redundant AI model clusters.
  - **System Errors**: Logic-based glitches (e.g., incorrect emotional classification or a broken timeline graph) are minimized through exhaustive unit, integration, and regression testing.
  - **System Failure**: Critical shutdowns are prevented through disaster recovery plans, daily off-site encrypted backups, and high-availability architecture.
- **Graceful Degradation**: In the event of a partial system failure (e.g., the video analysis module is offline for maintenance), the system is designed for "Graceful Degradation," ensuring that text and audio analysis remain fully functional.

3.2.3 Domain III: Safety and Ethical AI Governance

Safety in an affective computing environment refers to the prevention of psychological distress or harm resulting from the system’s output or data misuse.

3.2.3.1 Hazard Mitigation and Risk Severity

The system utilizes a "Hazard Severity Matrix" to categorize risks:

- **High Severity Hazards**: Biometric data breaches or "False-Negative Critical Alerts," where the system fails to detect and notify a user of a severe negative emotional trend (e.g., extreme distress).
- **Moderate Severity Hazards**: AI misinterpretations (e.g., labeling a neutral state as "angry") that could negatively influence a user's self-perception if presented without context.
- **Operational Hazards**: Inadequate user authentication leading to the exposure of sensitive mental-health notifications.

3.2.3.2 Safety Achievement and Damage Limitation

- **Hazard Avoidance**: We prevent accidents through strict identity verification (email/phone verification and bot detection) and automated content moderation. AI vision and NLP tools scan all media to filter out graphic, violent, or triggering content.
- **Interpretive Safety**: The system provides "Emotional Confidence Indicators." It explicitly disclaims that analysis results are interpretive and not clinical diagnoses, encouraging users to view the data as a tool for awareness rather than a definitive medical statement.
- **Incident Containment**: In the case of a data leak, the platform features an "Automatic Token Revocation" system to immediately terminate compromised sessions and limit potential damage.

3.2.4 Domain IV: Usability and User Experience (UX)

Usability focuses on making the complex technology of emotion detection accessible, intuitive, and efficient for all users.

3.2.4.1 Appearance, Interactivity, and Cognitive Load

- **Cognitive Load Reduction**: The dashboard is designed to present complex emotional data (heatmaps, timelines, pie charts) in a simplified, "User-Centric" format. Only essential tools—input, history, and alerts—are prioritized to prevent user overwhelm.
- **Information Architecture**: The system's categories are logically organized, allowing users to move from "Media Upload" to "Result Interpretation" in a single, streamlined sequence.

3.2.4.2 Consistency and Responsive Design

- **Platform Uniformity**: The platform maintains strict consistency in layout, icons, and color-coding (e.g., consistent colors for specific emotions) across both the Angular web portal and the Flutter mobile application.
- **Adaptive Layouts**: The UI is fully responsive, ensuring that emotion graphs and timeline visualizations remain readable and interactive across smartphones, tablets, and desktop displays.

3.2.5 Domain V: Efficiency and Performance Optimization

Efficiency measures the system's ability to provide rapid emotional insights despite the high computational demands of deep learning models.

3.2.5.1 Processing Throughput and Latency

- **Optimized Pipelines**: We utilize "Model Pruning" and "GPU Acceleration" for our Python microservices. This ensures that text and image analysis results are available in seconds, while video processing is optimized via parallel frame inference.
- **Cycle Time and Sprints**: Using Agile development (Sprints), we continuously monitor and reduce the "Cycle Time" for model recalibration, ensuring the AI becomes faster and

more accurate with every update.

3.2.5.2 Resource Management

- **Multimedia Compression**: Intelligent compression techniques are applied to images, audio, and video files to ensure fast uploads and reduced storage load without compromising the accuracy of the emotion recognition models.
- **Predictive Insights**: The system uses predictive algorithms to anticipate user emotional trends (e.g., detecting rising stress early), allowing it to pre-calculate certain summary reports.

3.2.6 Domain VI: Scalability and Maintainability

Scalability ensures the platform can grow with its user base, while maintainability ensures it can evolve.

3.2.6.1 Scaling Models

- **Horizontal vs. Vertical Scaling**: The platform primarily uses "Horizontal Scaling" (adding more server nodes) to handle increased volumes of text, audio, and video uploads. "Vertical Scaling" (upgrading CPU/GPU on existing nodes) is used for specific heavy-duty AI training tasks.
- **Auto-Scaling and Elasticity**: The system dynamically provisions additional resources during spikes in traffic (e.g., during high-traffic periods) and scales down during quieter hours to optimize operational costs.

3.2.6.2 Maintainability and Upgradability

- **Modularity**: A strict modular structure ensures that each component—text analysis, audio processing, image recognition, and the timeline engine—can be updated or replaced independently without disrupting the entire system.
- **Standardization and Testability**: Clean coding standards, unified naming conventions, and comprehensive API documentation facilitate long-term maintenance. Automated test cases verify that new updates do not cause "Regression Errors" in existing features.

3.2.7 Domain VII: Compatibility and Accessibility

Compatibility ensures universal access, while accessibility ensures inclusive access.

3.2.7.1 Cross-Platform and Backward Compatibility

- **Omni-Channel Support**: The system is tested across major browsers (Chrome, Firefox, Safari, Edge) and operating systems (Windows, macOS, Linux, Android, iOS).
- **Legacy Support**: "Backward Compatibility" ensures that older devices with limited processing power can still access lightweight emotional summaries and history logs through optimized UI elements.

3.2.7.2 Accessibility and WCAG Compliance

- **Inclusive Design**: The platform adheres to WCAG (Web Content Accessibility Guidelines). Features include high-contrast modes for users with color vision deficiencies, alt-text for all visual emotional indicators, and full keyboard navigation support.
- **Assistive Technology Integration**: The system provides "Text-to-Speech (TTS)" support, reading aloud emotional analysis results and recommendations for users with visual or cognitive impairments.

3.3 Tools and Methods in Our System

The "Emotion Detection with Timeline Analysis" system is engineered as a synergistic ecosystem of advanced technologies, integrated to move beyond the limitations of static emotion classification. The methodology is defined by a modular, microservices-based approach that ensures scalability, precision in temporal tracking, and a seamless user experience. By decoupling the intelligence layer from the orchestration and visualization layers, the system achieves high performance in processing high-density multimodal data.

The system's technical foundation rests upon four primary pillars:

1. **Distributed AI Intelligence:** Python-based microservices utilizing state-of-the-art Deep Learning models.
2. **Core Orchestration Layer:** An ASP.NET Core environment designed for data fusion and session management.
3. **Reactive Visualization Environment:** An Angular-driven interface optimized for dynamic data representation.
4. **Temporal Synthesis Methods:** Custom algorithms for cross-modal synchronization and longitudinal pattern detection.

3.3.1 Multimodal Artificial Intelligence Architecture

The intelligence layer is constructed using a series of independent Python microservices. This design choice facilitates the use of specialized libraries—such as PyTorch and TensorFlow—while allowing each modality to scale its computational resources according to the complexity of the input.

3.3.1.1 Linguistic Affective Analysis and Intensity Weighting

The text analysis service utilizes the j-hartmann/emotion-english-distilroberta-base transformer model. The methodology extends beyond simple classification by implementing a three-stage NLP pipeline. First, raw text undergoes **contextual segmentation**, where the input is divided into discrete semantic units (sentences). This allows the system to capture emotional shifts within a single narrative.

Second, the system applies a **Heuristic Intensity Weighting** algorithm. This custom logic identifies high-impact keywords (e.g., "devastated" vs. "sad") to adjust the model's confidence scores. Finally, a **Global-Local Fusion** approach reconciles individual sentence results with the overall sentiment of the text, ensuring that the final timeline reflects both immediate emotional spikes and the broader emotional context.

3.3.1.2 Computer Vision for Temporal Facial Expression Tracking

Facial analysis is achieved through the integration of **DeepFace** and **OpenCV**. The method involves high-frequency frame extraction, where video streams are sampled at optimized intervals (0.5 to 1.0 seconds) to balance detection accuracy with processing speed. Each extracted frame undergoes automated face detection and alignment to isolate the Region of Interest (ROI).

These frames are then processed through Convolutional Neural Networks (CNNs) to predict emotion probabilities. By mapping these predictions back to the original timestamps, the system constructs a high-resolution visual emotion curve that reveals micro-expressions and gradual transitions that static systems typically overlook.

3.3.1.3 Acoustic Signal Processing and Spectral Analysis

Vocal emotion detection focuses on the non-verbal cues of communication—pitch, energy, and rhythm. Utilizing the **librosa** library, the system extracts spectral features, including Mel-frequency cepstral coefficients (MFCCs). To maintain temporal alignment with other modalities, the audio stream is partitioned into short segments (typically 1–3 seconds). Each segment is analyzed to produce a localized emotional probability score, which is then serialized into the vocal-emotion timeline.

3.3.1.4 Multimodal Decision-Level Fusion Strategies

The core analytical challenge of the system is merging disparate data streams into a single unified timeline. Our system adopts a **Decision-Level Fusion** strategy. Instead of merging raw features, which is computationally prohibitive, the backend aggregates the probability vectors from each microservice. The fusion engine applies a **Confidence-Based Blending** method, where the modality with the highest model confidence and intensity weight for a specific timestamp takes precedence, creating a reliable and holistic emotional representation.

3.3.2 Enterprise Orchestration and Backend Integration

The backend serves as the "System Brain," providing the logical infrastructure required to transform raw AI outputs into a structured history of emotional intelligence.

3.3.2.1 ASP.NET Core Orchestration and API Gateway

The system utilizes **ASP.NET Core** to manage the lifecycle of an analysis session. The backend functions as an orchestration gateway, handling asynchronous requests to the Python microservices. When a user uploads a media file, the backend manages the file storage, triggers the appropriate AI services via RESTful calls, and remains idle until the JSON-formatted results are returned. This architecture ensures that the system can handle large files without blocking user interactions.

3.3.2.2 Relational Data Modeling for Longitudinal Tracking

To support the project's goal of long-term tracking, a robust relational database (SQL Server or PostgreSQL) is employed. The data model is optimized for the storage of **Vectorized Timelines**. Unlike standard records, these entries contain serialized arrays of timestamped data, allowing the system to reconstruct emotional graphs instantly. This persistence layer is the foundation for our "Mood History" feature, enabling the comparison of emotional patterns across multiple weeks or months.

3.3.3 Dynamic Visualization and Interaction Layer

The frontend is designed to make complex AI data interpretable through a modern, responsive, and data-centric user interface.

3.3.3.1 Angular Reactive Frontend Framework

Angular was selected for its sophisticated component-based architecture and its native support for **RxJS**. This allows the interface to handle the reactive state required for live analysis. Each segment of the dashboard—the video player, the live graph, and the emotion summary—is a decoupled component that updates in real-time as data streams arrive from the backend. **TailwindCSS** and **Angular Material** are used to provide a professional, accessible design system that reduces cognitive load for the user.

3.3.3.2 Affective Visualization and Interpretability

To bridge the gap between raw data and human understanding, the system integrates advanced charting libraries (Chart.js and D3.js). These tools generate several layers of visualization:

- **Temporal Line Graphs:** Representing the ebb and flow of valence and arousal over time.
- **Segmented Heatmaps:** Highlighting periods of intense emotional activity or "Peaks of Interest."
- **Aggregated Distribution Charts:** Providing a high-level summary of the session's dominant mood.

3.3.4 Temporal Synthesis and Timeline Generation

This section details the custom methodologies developed to fulfill the project's unique value proposition: the continuous emotional timeline.

3.3.4.1 Cross-Modal Synchronization Methods

The system implements a **Universal Time Axis** (UTA) to align data from different sources. Since text (sentences), audio (segments), and video (frames) have different temporal resolutions, the system uses a **Time-Alignment Algorithm** to interpolate and synchronize these inputs. This ensures that when a user hovers over a specific second in the video timeline, the system can display the corresponding facial expression, the spoken words, and the vocal tone simultaneously.

3.3.4.2 Active Alert Generation and Pattern Recognition

The backend includes a dedicated **Pattern Recognition Engine** that monitors for anomalies in the emotional data. By analyzing the frequency of "Negative Peaks" and the duration of sustained negative states (like high stress or sadness), the system generates "Active Alerts." These are not simple notifications but insights derived from historical data, warning the user of potentially concerning emotional trends.

3.3.5 Engineering Lifecycle and Quality Assurance

The development of this system followed an **Agile Scrum** methodology, emphasizing iterative refinement and technical robustness.

3.3.5.1 Modular Microservices and Scalability

By adopting a microservices architecture, the system is designed for future-proofing. New AI models or additional modalities (such as physiological data from wearables) can be integrated by simply deploying a new microservice and registering its endpoint in the .NET backend. This ensures the project remains a living platform capable of evolving with the field of Affective Computing.

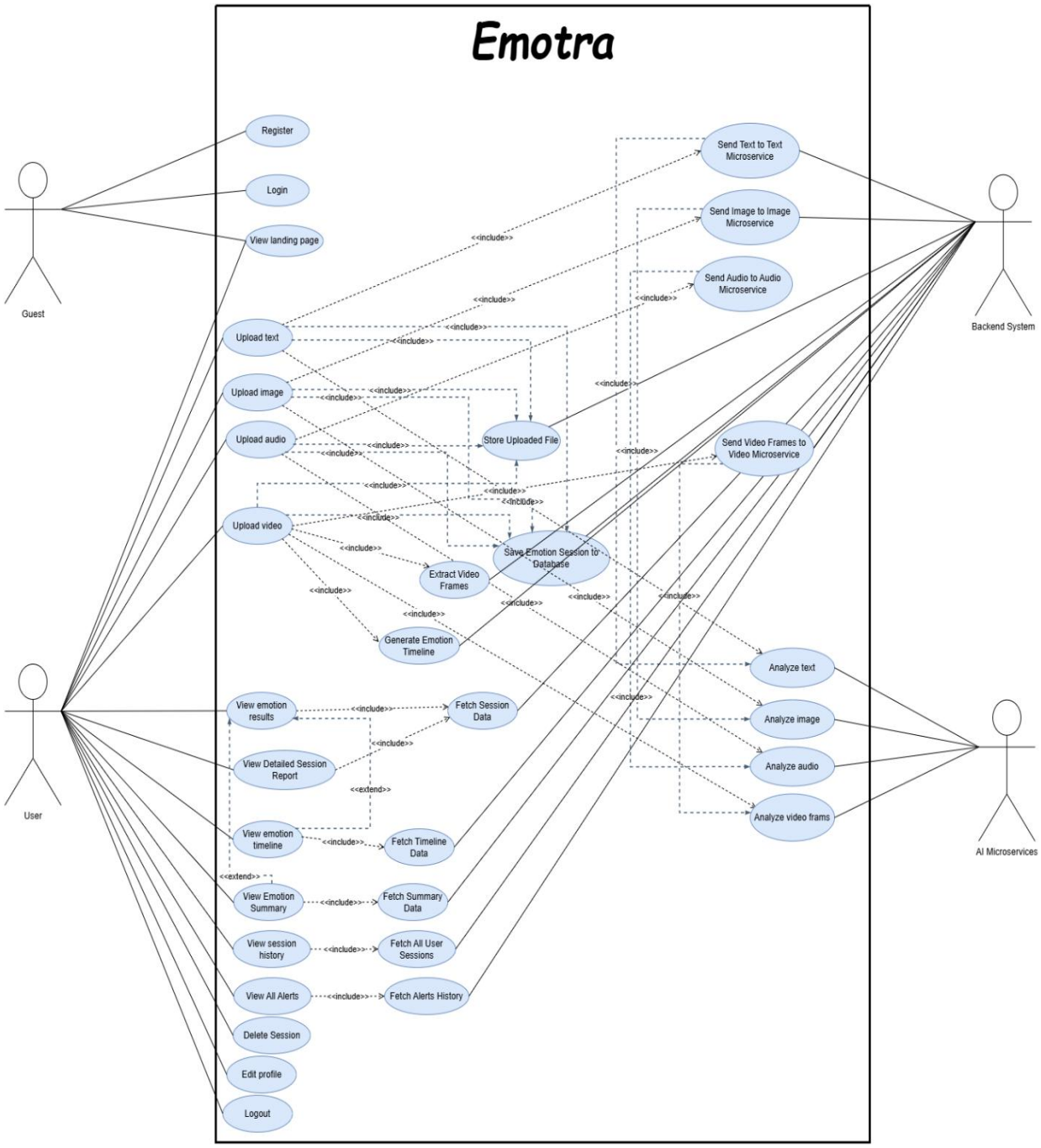
3.3.5.2 Validation and Testing Protocols

A comprehensive testing strategy was implemented to ensure system reliability:

- **Integration Testing:** Validating the RESTful handshake and JSON schema consistency between Python and .NET.
- **Temporal Accuracy Testing:** Ensuring that the AI-detected emotions align perfectly with the timestamps of the original media.
- **Performance Benchmarking:** Testing the system's response time under various loads, specifically focusing on the optimization of video frame extraction to maintain high throughput.

3.4. Diagrams

3.4.1. Use Case Diagram



Use Case Tables

Use Case#1	Register
Primary Actor	Visitor
Pre-Condition	The visitor opens the website and chooses to create a new account. The visitor must provide the required information (name, email, password) and accept the terms.
Main Scenario	➢ The visitor enters name, email, and password ➢ The system validates the email format and password strength ➢ The system checks if the email already exists ➢ If all information is valid, the system creates the account ➢ The visitor can now log in using the new credentials
Alternative Scenarios	➢ Invalid or missing data → system displays an error message ➢ Weak password → system requests a stronger password ➢ Email already registered → visitor must use another email

Use Case#2	Login
Primary Actor	Visitor
Pre-Condition	The visitor has already created an account on the website. The visitor must enter a valid email and password in the login form.
Main Scenario	➢ The visitor opens the login page. ➢ The visitor enters the registered email and password. ➢ The system verifies that the email exists in the database. ➢ The system checks if the entered password matches the stored password. ➢ If the credentials are correct, the system grants access and the visitor becomes a logged-in user.
Alternative Scenarios	➢ Incorrect email or password → the system displays an error message. ➢ Email not registered → the visitor must register before logging in. ➢ Required fields left empty → the system prompts the visitor to complete missing information.

Use Case#3	View Landing Page
Primary Actor	Visitor
Pre-Condition	The visitor opens the website without being logged in.
Main Scenario	➢ The visitor enters the website URL. ➢ The system loads the public landing page. ➢ The page displays general information about the platform and its features. ➢ The visitor can choose to register or log in from the landing page.
Alternative Scenarios	➢ The landing page fails to load due to a network issue; the system displays a connection error. ➢ System maintenance prevents displaying the landing page; a maintenance message is shown.

Use Case#4	View Landing Page
Primary Actor	User
Pre-Condition	The user is already logged in to the system.
Main Scenario	➢ The user opens the home or landing page after logging in. ➢ The system loads the user-specific landing page. ➢ The page displays the user dashboard, recent sessions, and navigation options. ➢ The user can move to upload pages, view analysis history, or access profile settings.
Alternative Scenarios	➢ The landing page fails to load due to a network issue; the system displays a connection error. ➢ System maintenance is in progress; a maintenance message is shown.

Use Case#5	Upload Text
Primary Actor	User
Pre-Condition	The user is logged in. The user accesses the upload page and chooses to upload a text input. The text must be provided in a supported format.
Main Scenario	➢ The user opens the text upload section. ➢ The user enters or pastes the text into the provided field. ➢ The system receives the text and stores it in the backend. ➢ The system sends the text to the text emotion analysis microservice. ➢ The AI processes the text and returns the emotion results. ➢ The system saves the final emotion session to the user's history.

Alternative Scenarios	➢ The text field is empty; the system requests the user to provide valid text. ➢ The text exceeds the allowed size; the system asks the user to shorten the input. ➢ The AI service fails to respond; the system displays an error and asks the user to try again later.

Use Case#6	**Upload Image**
Primary Actor	User
Pre-Condition	The user is logged in. The user opens the upload page and selects the option to upload an image file. The image must be in a supported format such as JPG or PNG.
Main Scenario	➢ The user chooses an image from their device. ➢ The system receives the uploaded file and stores it in the backend. ➢ The system sends the image to the image emotion analysis microservice. ➢ The AI analyzes the facial expression in the image and returns the emotion results. ➢ The system saves the final emotion session to the user's history.
Alternative Scenarios	➢ Unsupported file type; the system notifies the user to upload a valid image format. ➢ Image is corrupted or unreadable; the system asks the user to try another file. ➢ The AI service fails to respond; the system displays an error message and suggests retrying later.

Use Case#7	Upload Audio
Primary Actor	User
Pre-Condition	The user is logged in. The user navigates to the upload page and selects the option to upload an audio file. The audio must be in a supported format such as WAV or MP3.
Main Scenario	➢ The user selects an audio file from their device. ➢ The system receives the file and stores it in the backend. ➢ The system sends the audio to the audio emotion analysis microservice. ➢ The AI analyzes the audio and returns the detected emotions. ➢ The system saves the final emotion session to the user's history.
Alternative Scenarios	➢ Unsupported or corrupted audio file; the system requests the user to provide a valid file. ➢ The file exceeds the maximum size limit; the system notifies the user to upload a smaller file. ➢ The AI microservice is unavailable; the system displays an error and asks the user to try again later.

Use Case#8	Upload Video
Primary Actor	User
Pre-Condition	The user is logged in. The user opens the upload page and selects the option to upload a video file. The video must be in a supported format such as MP4 or MOV.
Main Scenario	➢ The user selects a video file from their device. ➢ The system receives the file and stores it in the backend. ➢ The system extracts frames from the video for analysis. ➢ The system sends the extracted frames to the video emotion analysis microservice. ➢ The AI analyzes each frame and returns emotion results across the timeline. ➢ The system generates the final emotion timeline based on the frame-level data. ➢ The system saves the completed emotion session to the user's history.
Alternative Scenarios	➢ Unsupported video format; the system requests the user to upload a valid format. ➢ Video is corrupted or cannot be processed; the system asks the user to try another file. ➢ Video file exceeds size limits; the system informs the user to upload a smaller file.

	➢ AI service or frame extraction fails; the system displays an error and asks the user to try again later.

Use Case#9	View Emotion Results
Primary Actor	User
Pre-Condition	The user is logged in. The user has previously uploaded text, image, audio, or video and has at least one completed emotion analysis session saved in the system.
Main Scenario	➢ The user opens the analysis results page or selects a completed session. ➢ The system retrieves the stored emotion analysis data from the backend. ➢ The system displays the final detected emotion and its confidence level. ➢ The user can switch between different sections such as summary, timeline, or detailed view.
Alternative Scenarios	➢ No completed sessions exist; the system informs the user that no results are available. ➢ The session data cannot be retrieved due to a backend error; the system displays a retrieval error message. ➢ The saved session is corrupted or incomplete; the system notifies the user that this analysis cannot be displayed.

Use Case#10	View Emotion Timeline
Primary Actor	User
Pre-Condition	The user is logged in. A completed emotion analysis session exists that contains timeline data (audio segments or video frames). The user chooses to view the timeline of a selected session.
Main Scenario	➢ The user opens the session and selects the option to view the emotion timeline. ➢ The system retrieves the stored timeline data from the backend. ➢ The system loads the timeline graph and displays emotion changes over time. ➢ The user can move along the timeline to see how emotions shift throughout the session.
Alternative Scenarios	➢ The selected session has no timeline data; the system informs the user that timeline view is unavailable. ➢ Backend retrieval fails; the system displays an error and asks the user to try again later. ➢ Timeline data is corrupted or incomplete; the system notifies the user that the timeline cannot be displayed.

Use Case#10	View Emotion Timeline
Primary Actor	User
Pre-Condition	The user is logged in. A completed emotion analysis session exists that contains timeline data (audio segments or video frames). The user chooses to view the timeline of a selected session.
Main Scenario	➢ The user opens the session and selects the option to view the emotion timeline. ➢ The system retrieves the stored timeline data from the backend. ➢ The system loads the timeline graph and displays emotion changes over time. ➢ The user can move along the timeline to see how emotions shift throughout the session.
Alternative Scenarios	➢ The selected session has no timeline data; the system informs the user that timeline view is unavailable. ➢ Backend retrieval fails; the system displays an error and asks the user to try again later. ➢ Timeline data is corrupted or incomplete; the system notifies the user that the timeline cannot be displayed.

Use Case#11	View Emotion Summary
Primary Actor	User
Pre-Condition	The user is logged in. A completed emotion analysis session exists with stored summary data (overall emotion distribution, dominant emotion, confidence levels). The user selects the option to view the summary of a session.
Main Scenario	➢ The user opens the analysis session and chooses the "Emotion Summary" view. ➢ The system retrieves the stored summary data from the backend. ➢ The system displays the overall emotion distribution using charts or visual indicators. ➢ The user views key insights such as dominant emotion, confidence percentages, and positive/negative/neutral ratios.
Alternative Scenarios	➢ The selected session does not contain summary data; the system notifies the user. ➢ Backend retrieval fails; the system displays an error and suggests trying again later. ➢ Summary data is corrupted or incomplete; the system informs the user that the summary cannot be shown.

Use Case#12	View Session History
Primary Actor	User
Pre-Condition	The user is logged in. The user has at least one saved emotion analysis session in their history.
Main Scenario	➢ The user navigates to the "Session History" page. ➢ The system retrieves all saved analysis sessions associated with the user. ➢ The system displays the list of sessions, including dates and session types. ➢ The user can select any session to view its detailed results.
Alternative Scenarios	➢ The user has no previous sessions; the system displays a message indicating the history is empty. ➢ The backend fails to retrieve the session list; the system shows an error message. ➢ Some sessions cannot be loaded due to corruption; the system skips them and informs the user if needed.

Use Case#13	View Detailed Session Report
Primary Actor	User
Pre-Condition	The user is logged in.

	The user has selected an existing session from the session history.
Main Scenario	➢ The user clicks on a specific analysis session. ➢ The system retrieves all stored data for the selected session from the backend. ➢ The system displays the full session report, including the final emotion, confidence values, summary, and timeline preview. ➢ The user can navigate between different sections of the detailed report.
Alternative Scenarios	➢ The selected session cannot be found; the system informs the user that the session is unavailable. ➢ Session data retrieval fails due to a system error; an error message is shown. ➢ Some parts of the session data are incomplete; the system displays available information and warns the user about missing data.

Use Case#14	**View All Alerts**
Primary Actor	User
Pre-Condition	The user is logged in. The system has previously generated one or more emotion alerts based on negative emotional patterns.
Main Scenario	➢ The user navigates to the “Alerts” section from the dashboard or menu. ➢ The system retrieves all stored alerts associated with the user from the backend. ➢ The system displays the list of alerts, including the alert message, date, and type. ➢ The user can open any alert to read additional details if available.
Alternative Scenarios	➢ No alerts are available; the system displays a message indicating there are no alerts. ➢ The backend fails to retrieve the alerts; the system shows an error message. ➢ Some alerts are corrupted or unreadable; the system skips them and informs the user if necessary.

Use Case#15	**Receive Negative Emotion Alert**
Primary Actor	User
Pre-Condition	The user is logged in. A completed emotion analysis session contains consistent or repeated negative emotion patterns. The backend system has generated an alert based on these patterns.
Main Scenario	➢ The system analyzes the user’s recent emotion sessions and detects negative patterns.

	➢ The backend generates an alert and stores it in the user's alert history. ➢ The user receives the new alert in the alerts section or as a notification within the system. ➢ The user opens the alert to review the details and recommended actions (if provided).
Alternative Scenarios	➢ The alert cannot be generated due to a backend error; no notification is created. ➢ The user tries to open the alert but data retrieval fails; the system displays an error message. ➢ The alert details are incomplete; the system shows the available information and notifies the user of missing data.

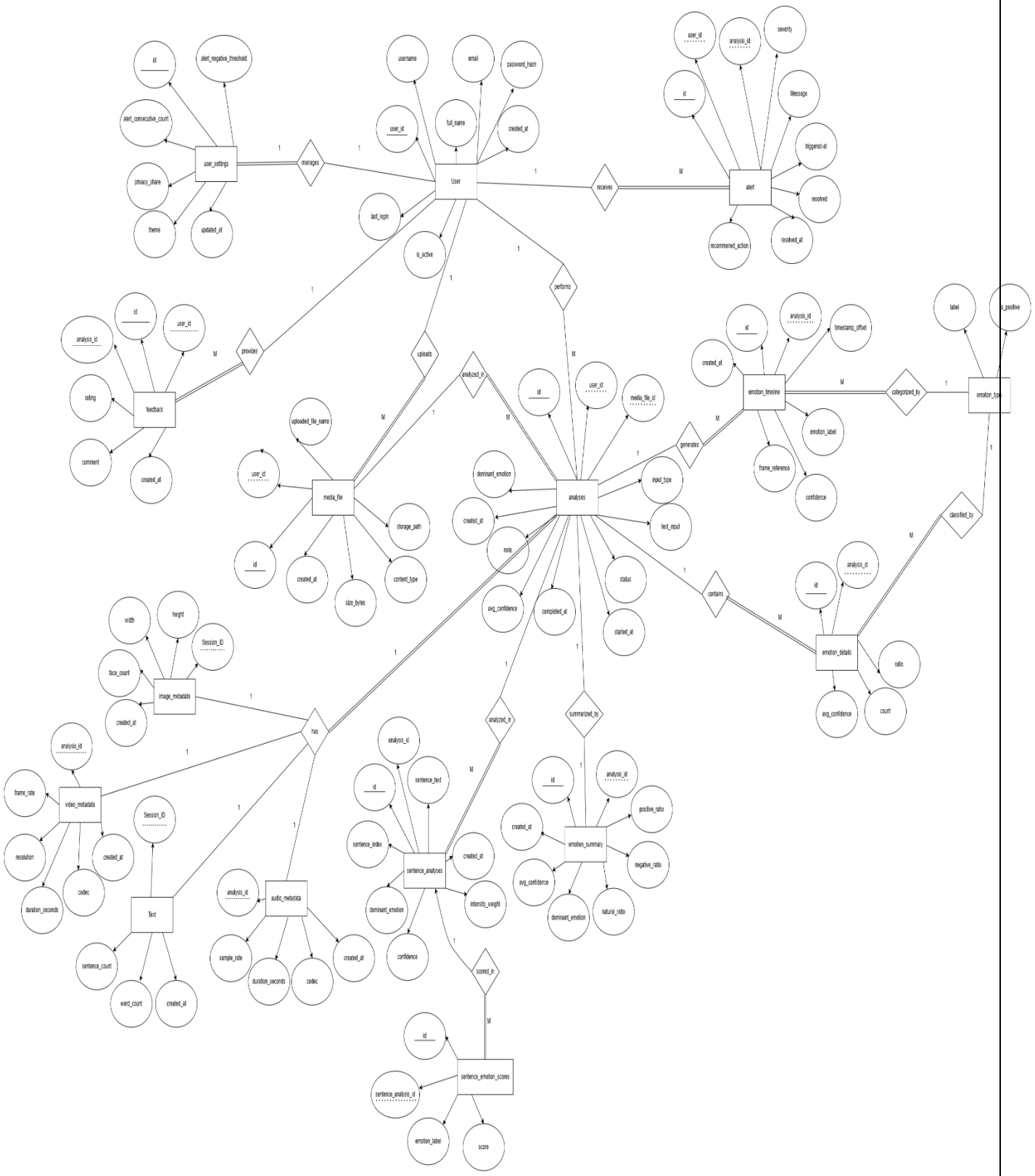
Use Case#16	**Delete Session**
Primary Actor	User
Pre-Condition	The user is logged in. The user has at least one saved analysis session in their session history. The user selects a specific session they want to delete.
Main Scenario	➢ The user opens the session history page. ➢ The user selects a session and chooses the "Delete" option. ➢ The system confirms the delete action to avoid accidental removal. ➢ The system removes the session from the database. ➢ The session is no longer visible in the user's session history.
Alternative Scenarios	➢ The user cancels the delete confirmation; the session remains unchanged. ➢ The session cannot be deleted due to a backend error; the system shows an error message. ➢ The selected session does not exist or has already been deleted; the system notifies the user.

Use Case#17	**Edit Profile**
Primary Actor	User
Pre-Condition	The user is logged in. The user opens the profile settings page. The user has access to editable profile fields such as name, email (if allowed), or profile picture.
Main Scenario	➢ The user navigates to the profile settings page. ➢ The system displays the current profile information. ➢ The user updates one or more fields (for example: name, password, profile picture). ➢ The system validates the updated information. ➢ The system saves the new profile data to the database. ➢ The user sees a confirmation message indicating that the profile was updated successfully.

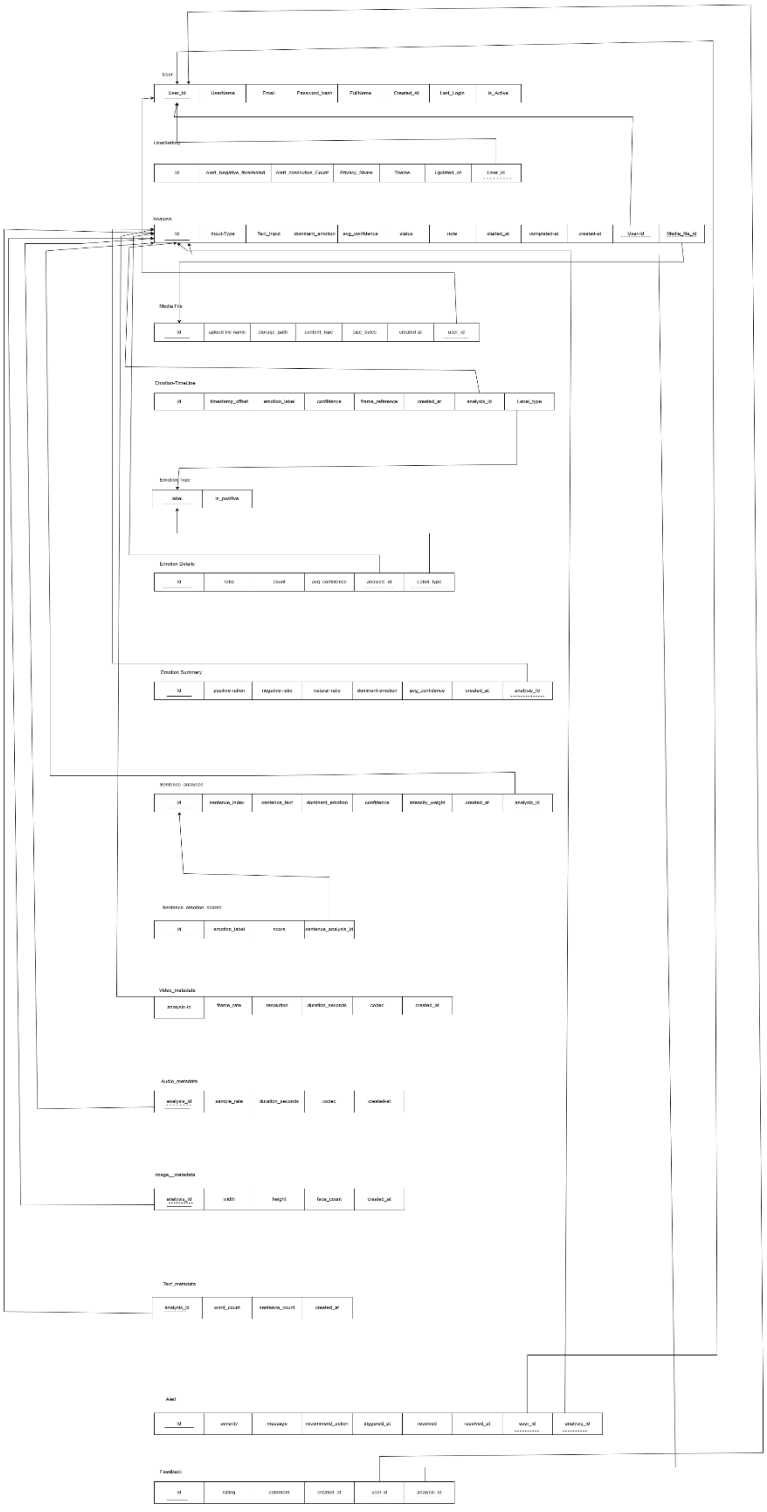
Alternative Scenarios	➢ Required fields are left empty; the system displays a validation error. ➢ The user enters an invalid email format (if editable); the system requests correction. ➢ The password does not meet security requirements; the system rejects the update. ➢ The backend fails to save the update; the system displays an error message.

Use Case#18	**Logout**
Primary Actor	User
Pre-Condition	The user is logged in. The user chooses to end the current session.
Main Scenario	➢ The user opens the menu or settings and selects the "Logout" option. ➢ The system terminates the user's active session. ➢ The system clears authentication tokens or session data. ➢ The system redirects the user to the landing page or login page. ➢ The user is no longer authenticated in the system.
Alternative Scenarios	➢ A network or system error occurs during logout; the system displays an error message. ➢ The user's session has already expired; the system simply returns the user to the login page.

3.4.2 ERD

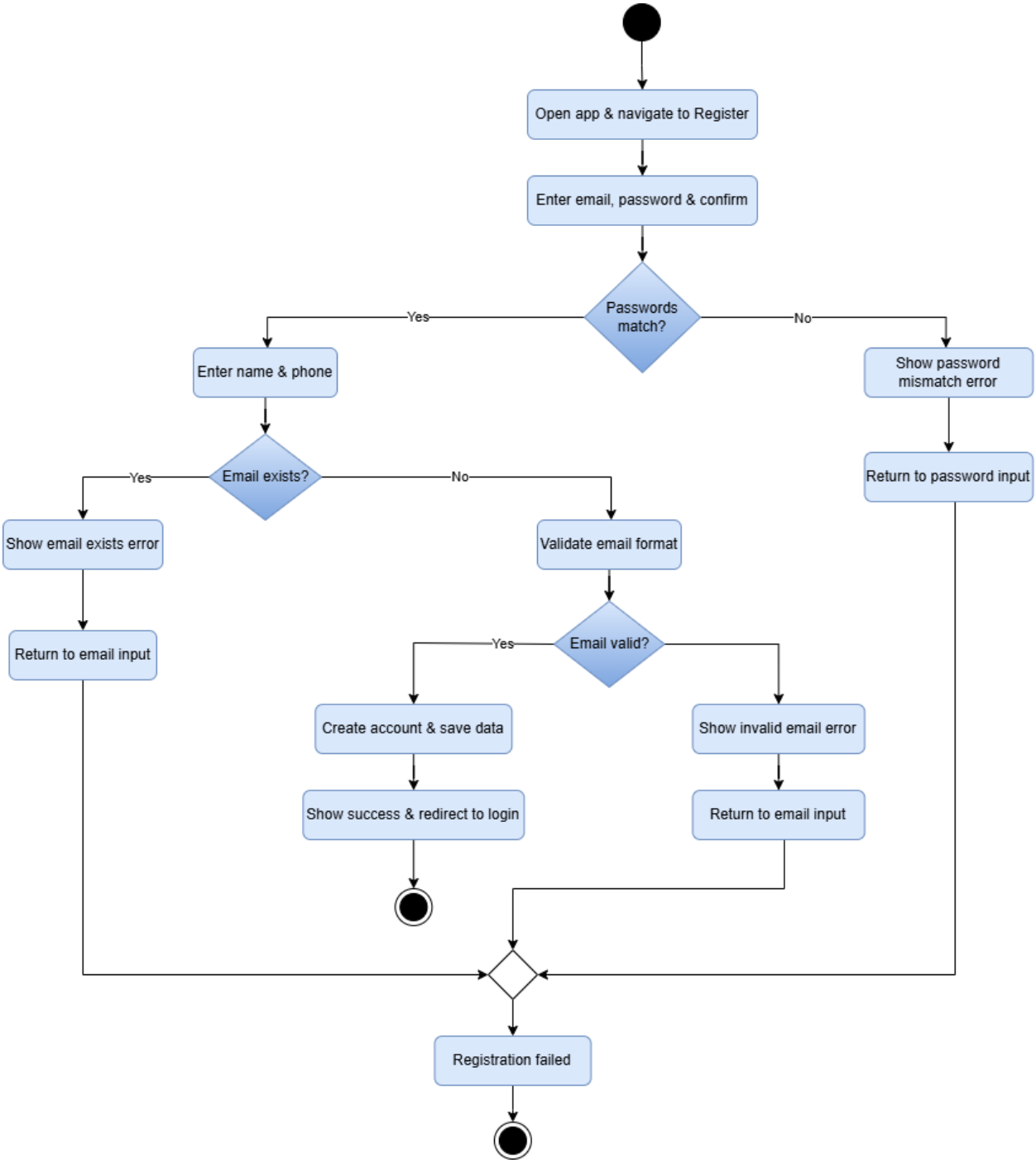

id
alert_negative_threshold
alert_consecutive_count
user_settings
privacy_share
theme
updated_at
1
manages
1
username
email
password_hash
user_id
full_name
created_at
User
last_login
is_active
1
receives
M
user_id
analysis_id
severity
id
Message
triggered-at
resolved
recommened_action
resolved_at
1
performs
M
1
provides
M
id
user_id
analysis_id
rating
comment
created_at
feedback
1
uploads
M
uploaded_file_name
user_id
id
created_at
size_bytes
content_type
storage_path
media_file
M
analyzed_in
M
id
user_id
media_file_id
dominant_emotion
created_at
note
avg_confidence
completed_at
status
started_at
analyses
1
generates
M
input_type
text_input
id
created_at
analysis_id
timestamp_offset
emotion_timeline
M
categorized_by
1
label
is_positive
emotion_type
1
emotion_label
frame_reference
confidence
classified_by
M
contains
M
id
analysis_id
emotion_details
ratio
avg_confidence
count
1
has
1
width
height
Session_ID
face_count
created_at
image_metadata
1
analyzed_in
M
summarized_by
1
analysis_id
frame_rate
resolution
codec
duration_seconds
created_at
video_metadata
1
Session_ID
Text
sentence_count
word_count
created_at
1
analysis_id
audio_metadata
sample_rate
duration_seconds
codec
created_at
analysis_id
id
sentence_index
sentence_text
sentence_analyses
dominant_emotion
confidence
created_at
intensity_weight
1
scored_in
M
id
analysis_id
created_at
emotion_summary
avg_confidence
dominant_emotion
natural_ratio
negative_ratio
positive_ratio
id
sentence_analysis_id
emotion_label
score
sentence_emotion_scores

3.4.3 Mapping

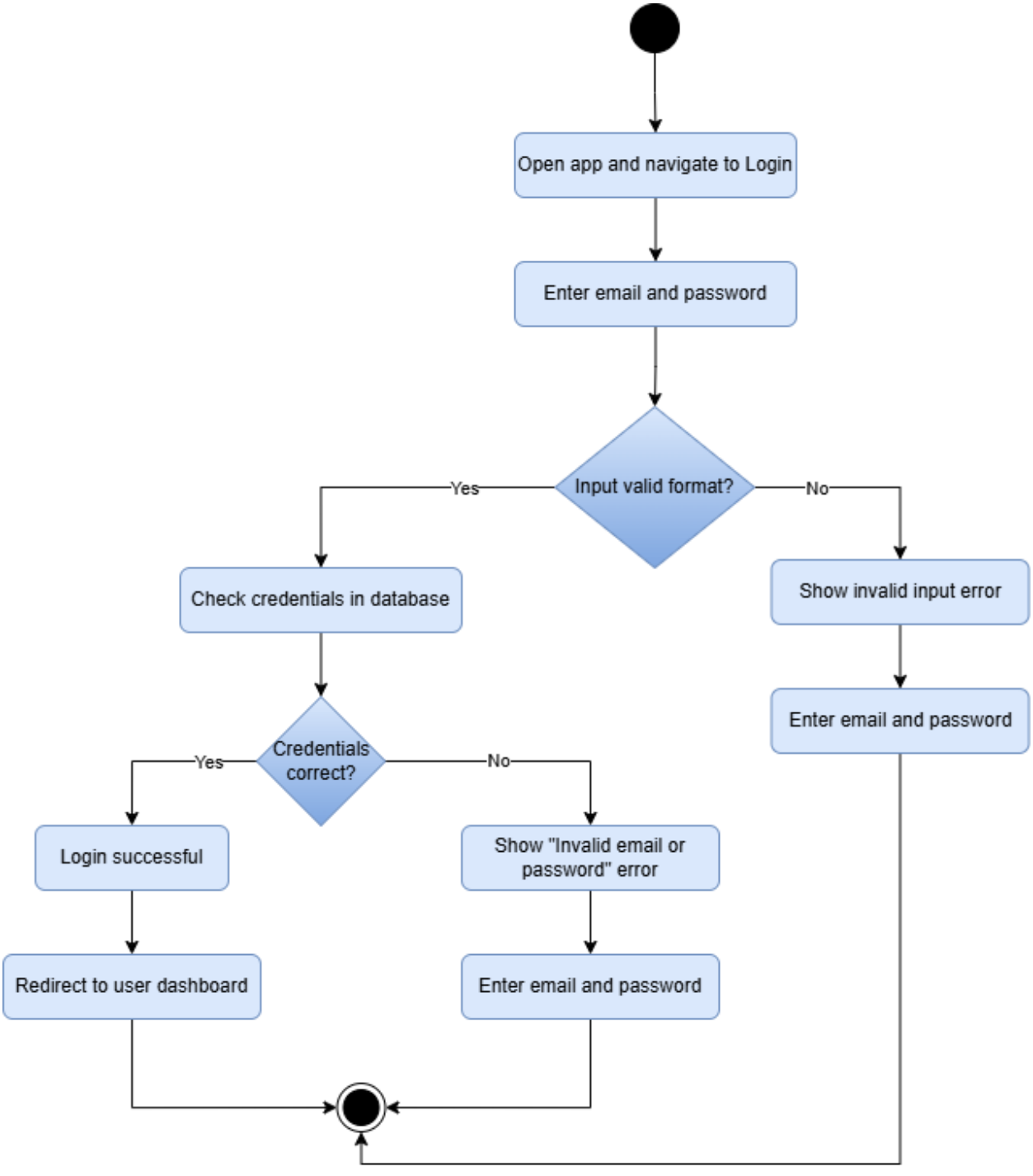


3.4.4. Activity Diagram

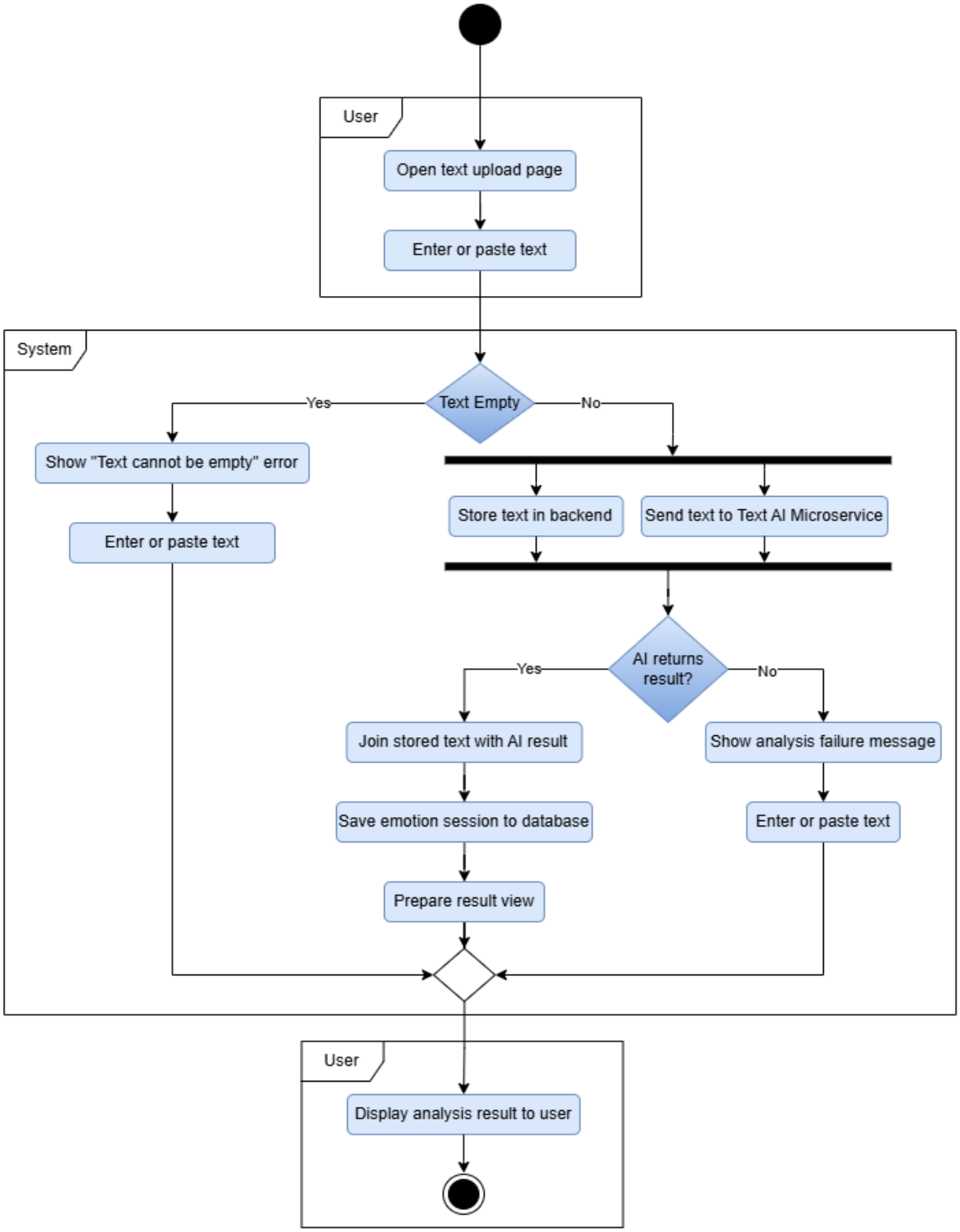
3.4.4.1. Register



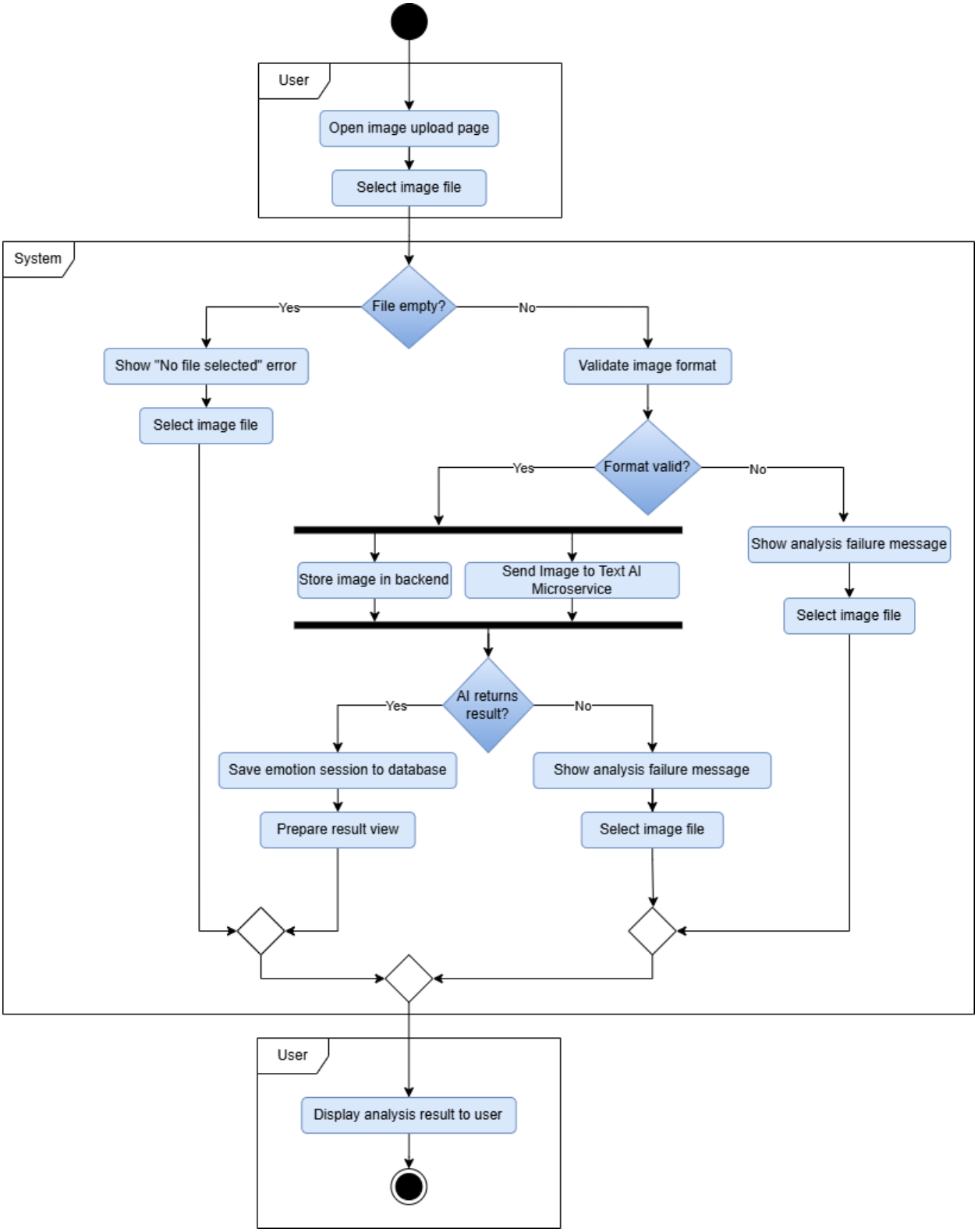
3.4.4.2. Login


Open app and navigate to Login
Enter email and password
Input valid format?
Yes
No
Check credentials in database
Show invalid input error
Enter email and password
Credentials correct?
Yes
No
Login successful
Show "Invalid email or password" error
Redirect to user dashboard
Enter email and password

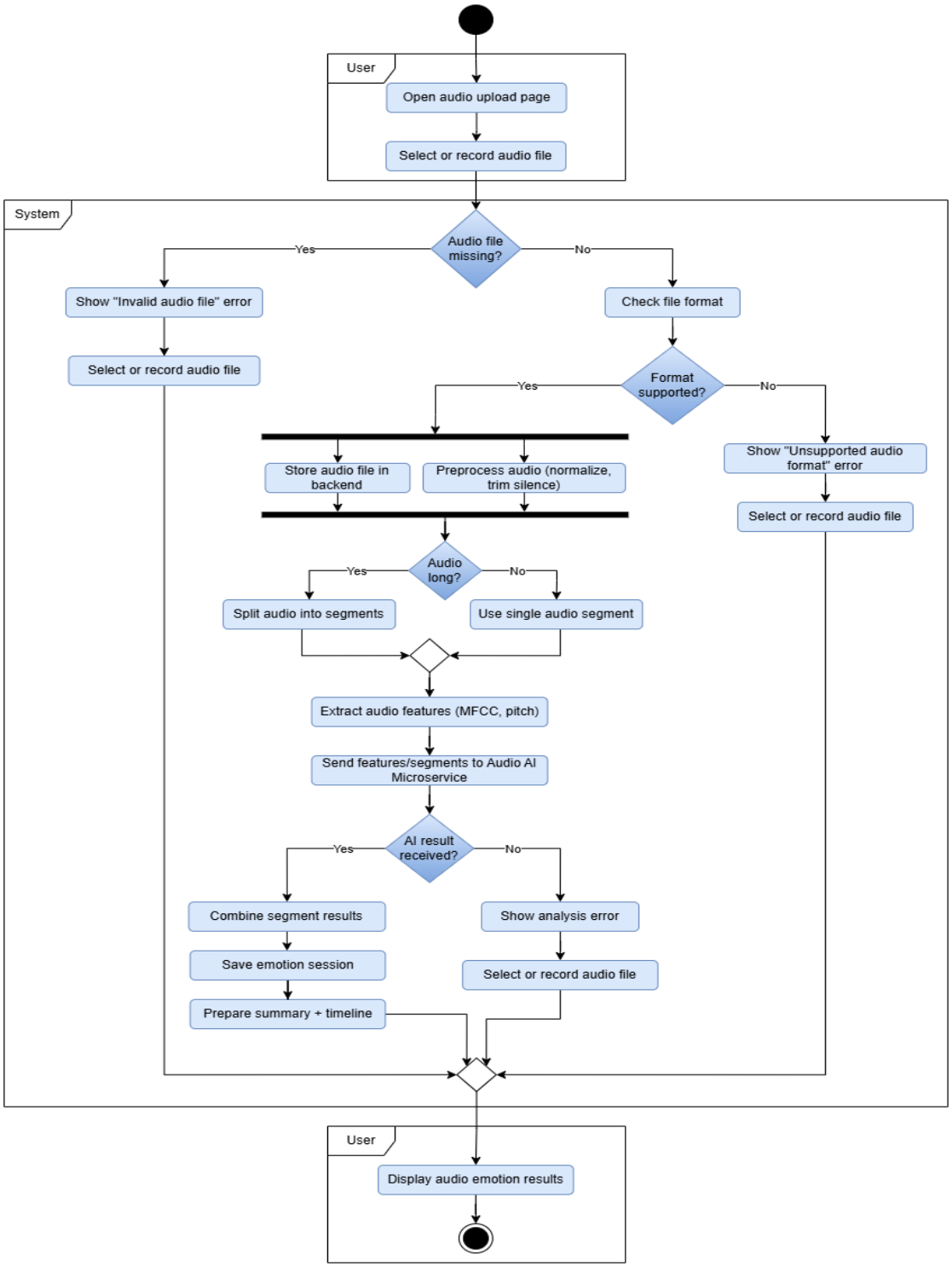
3.4.4.3. Upload Text



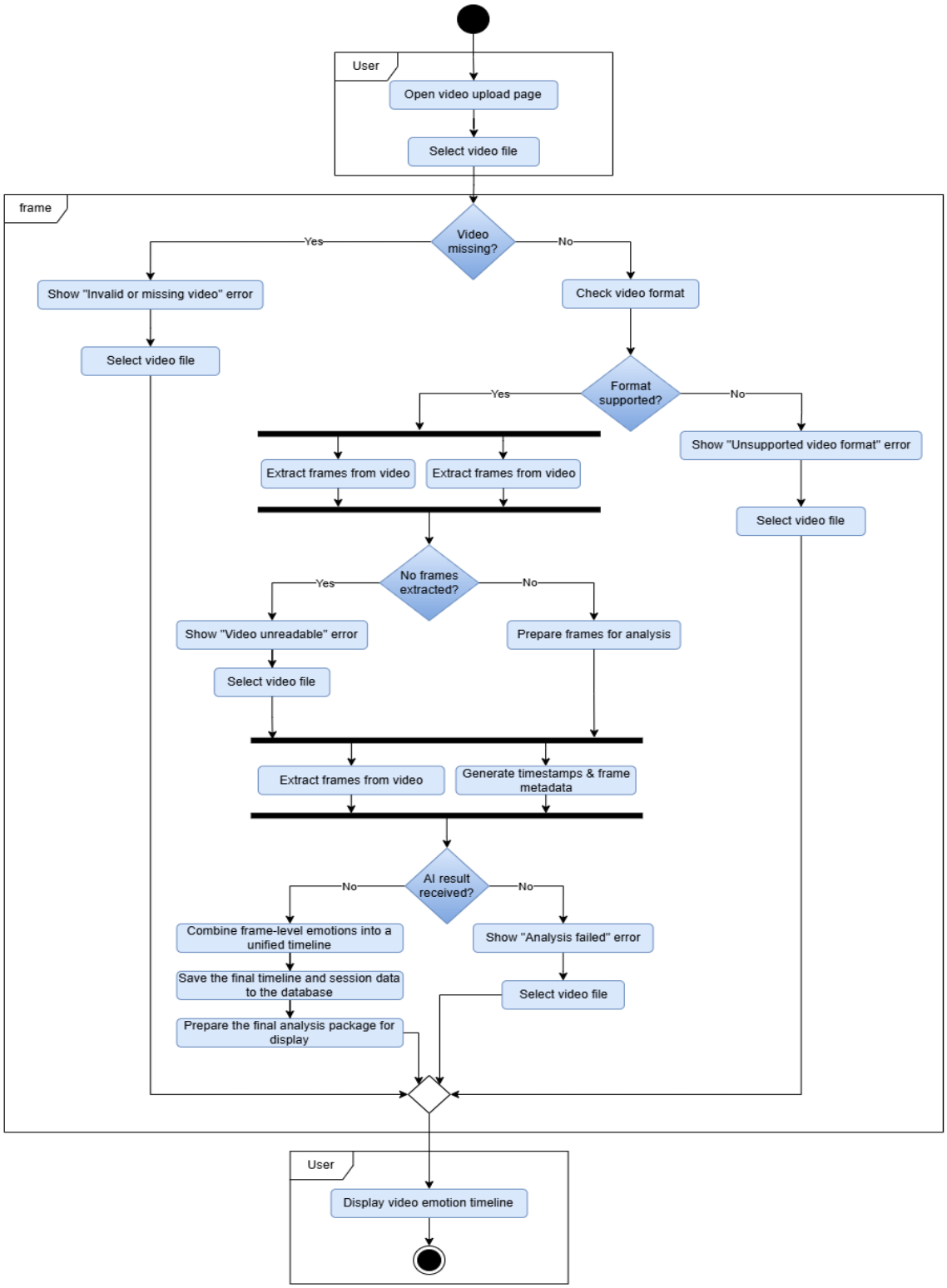
3.4.4.4. Upload Image



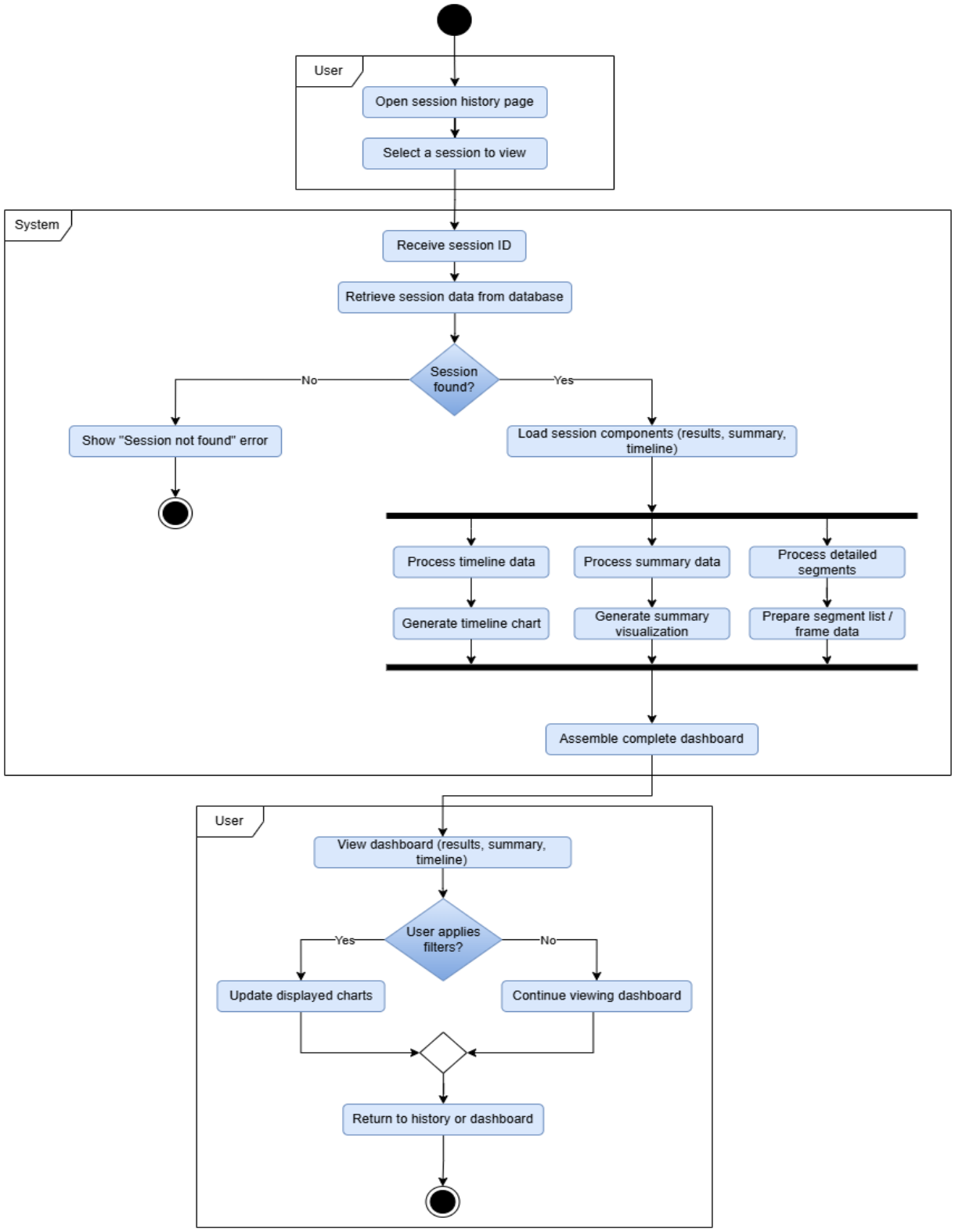
3.4.4.5. Upload Audio



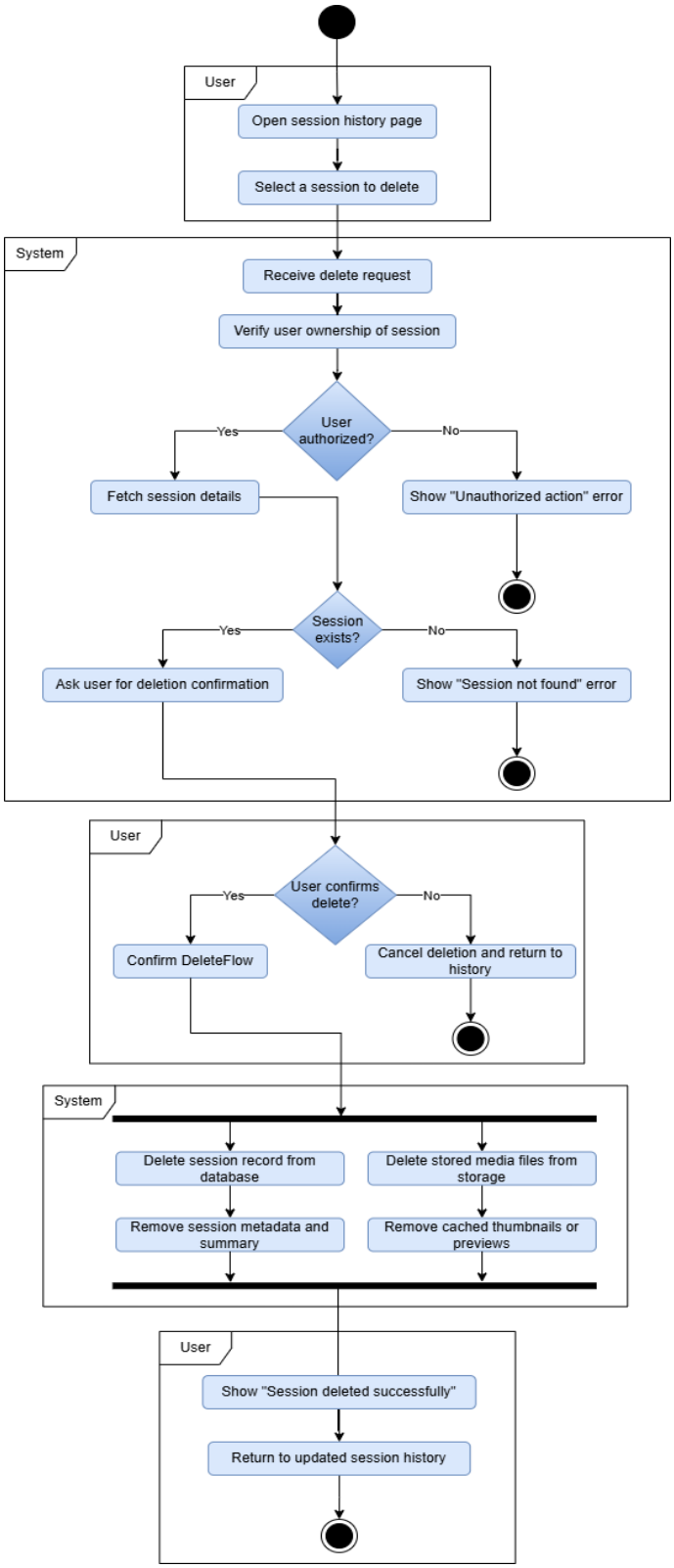
3.4.4.6. Upload Video



3.4.4.7. Upload View Session Dashboard

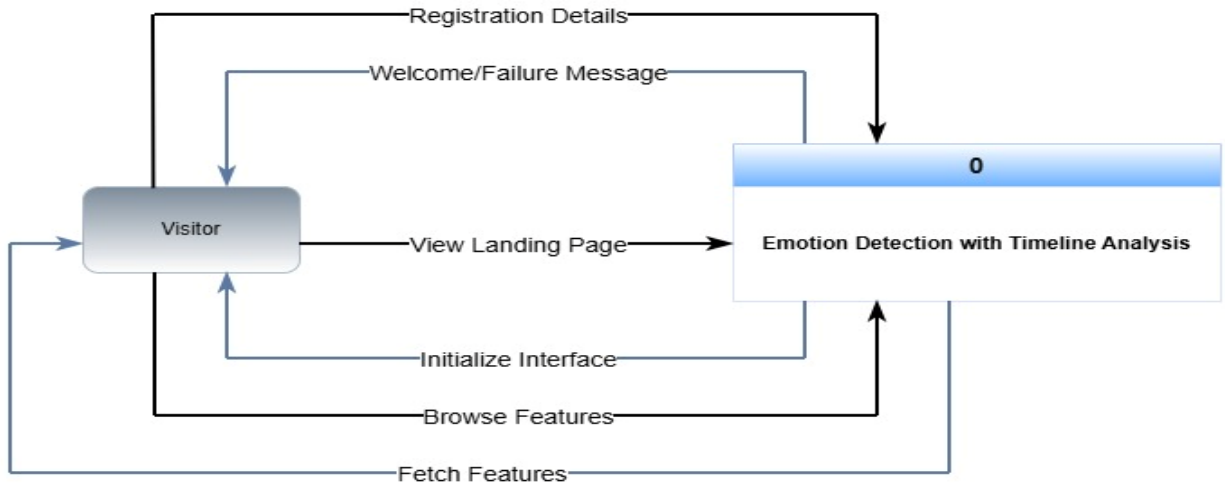


3.4.4.8. Delete Session

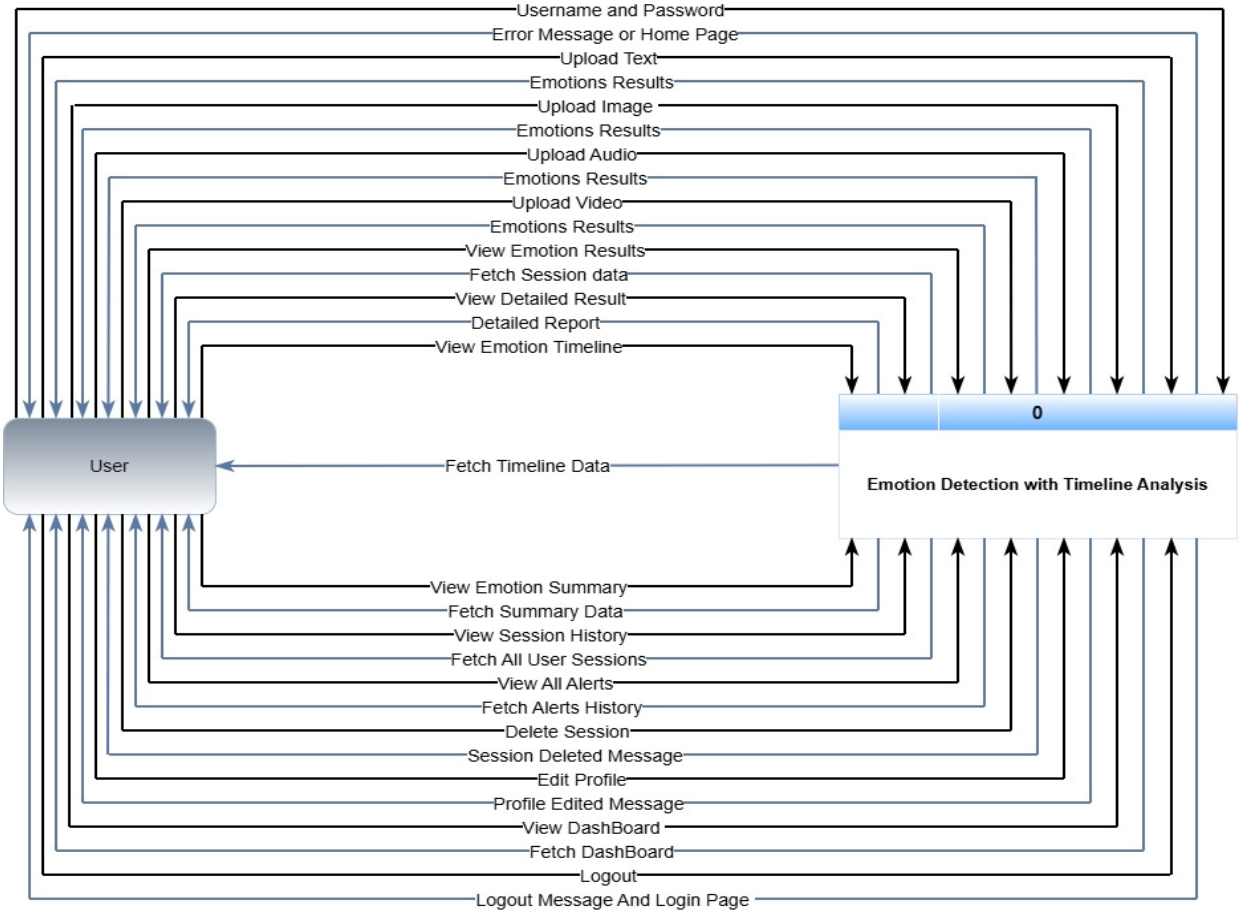


3.4.5. Context Diagram

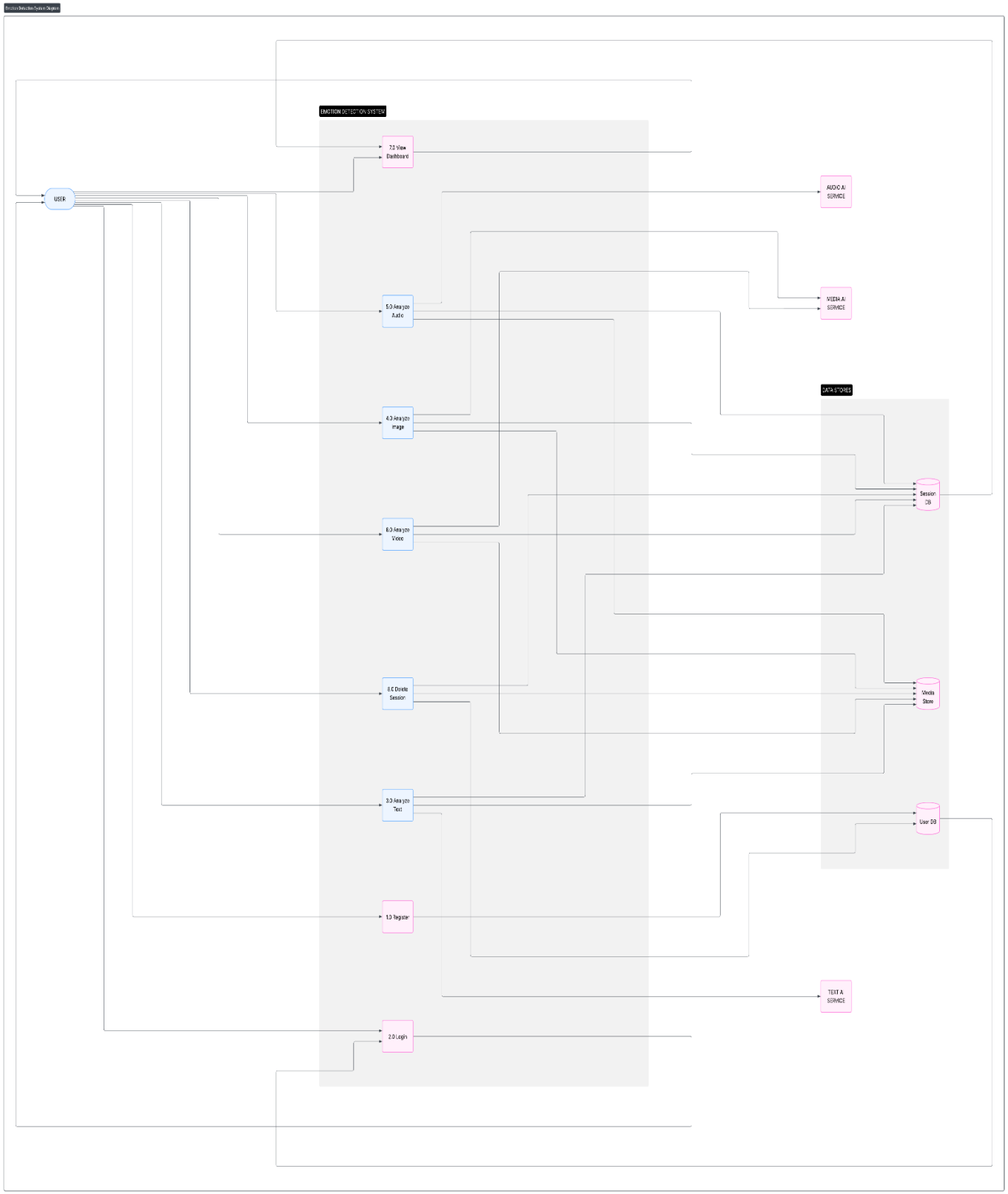
1. Visitor



2. User

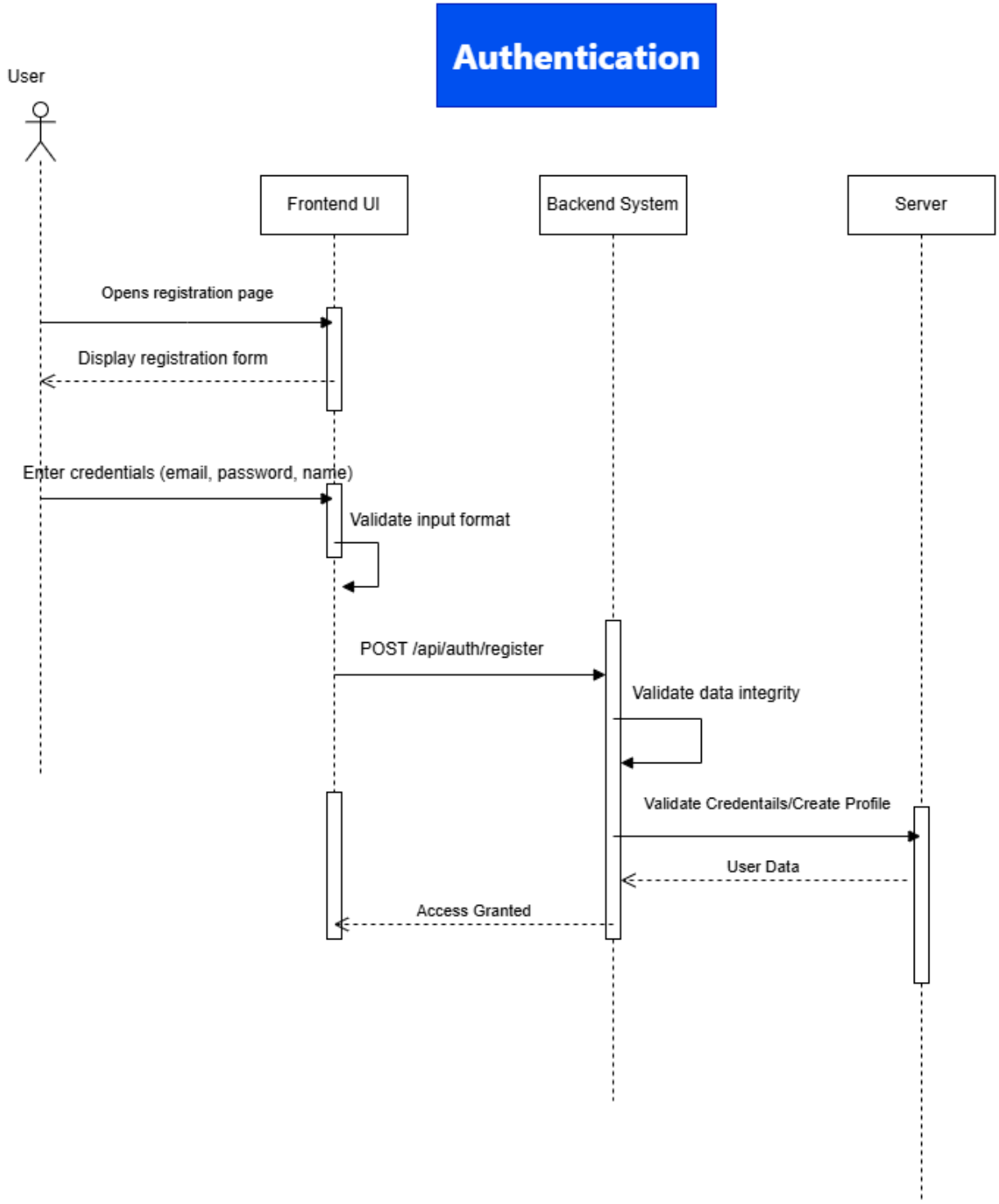


3.4.6. DFD

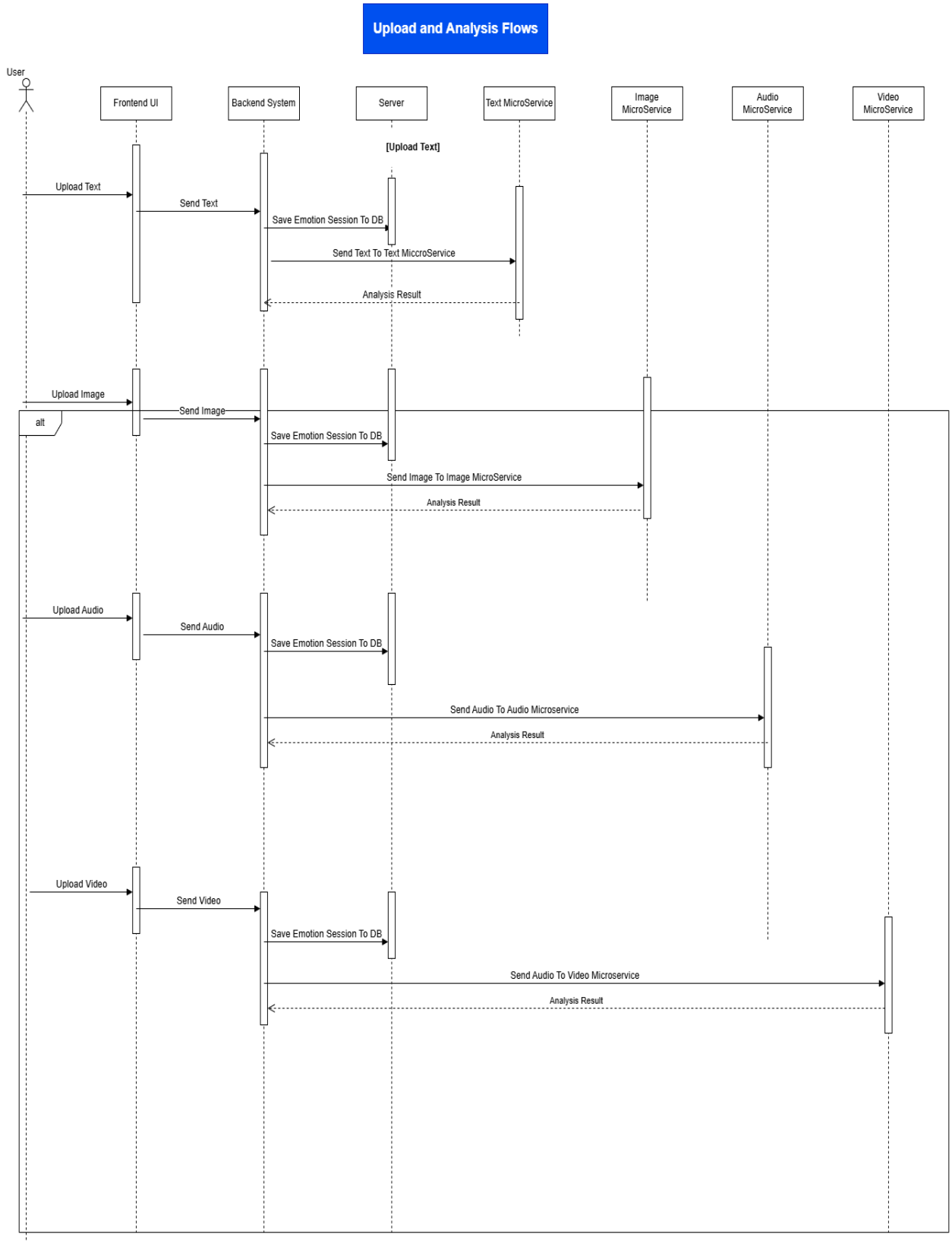

EMOTION DETECTION SYSTEM
USER
7.0 View Dashboard
5.0 Analyze Audio
4.0 Analyze Image
6.0 Analyze Voice
8.0 Delete Session
3.0 Analyze Text
1.0 Register
2.0 Login
AUDIO AI SERVICE
MEDIA AI SERVICE
TEXT AI SERVICE
DATA STORES
Session DB
Media Store
User DB

3.4.7. Sequence Diagram

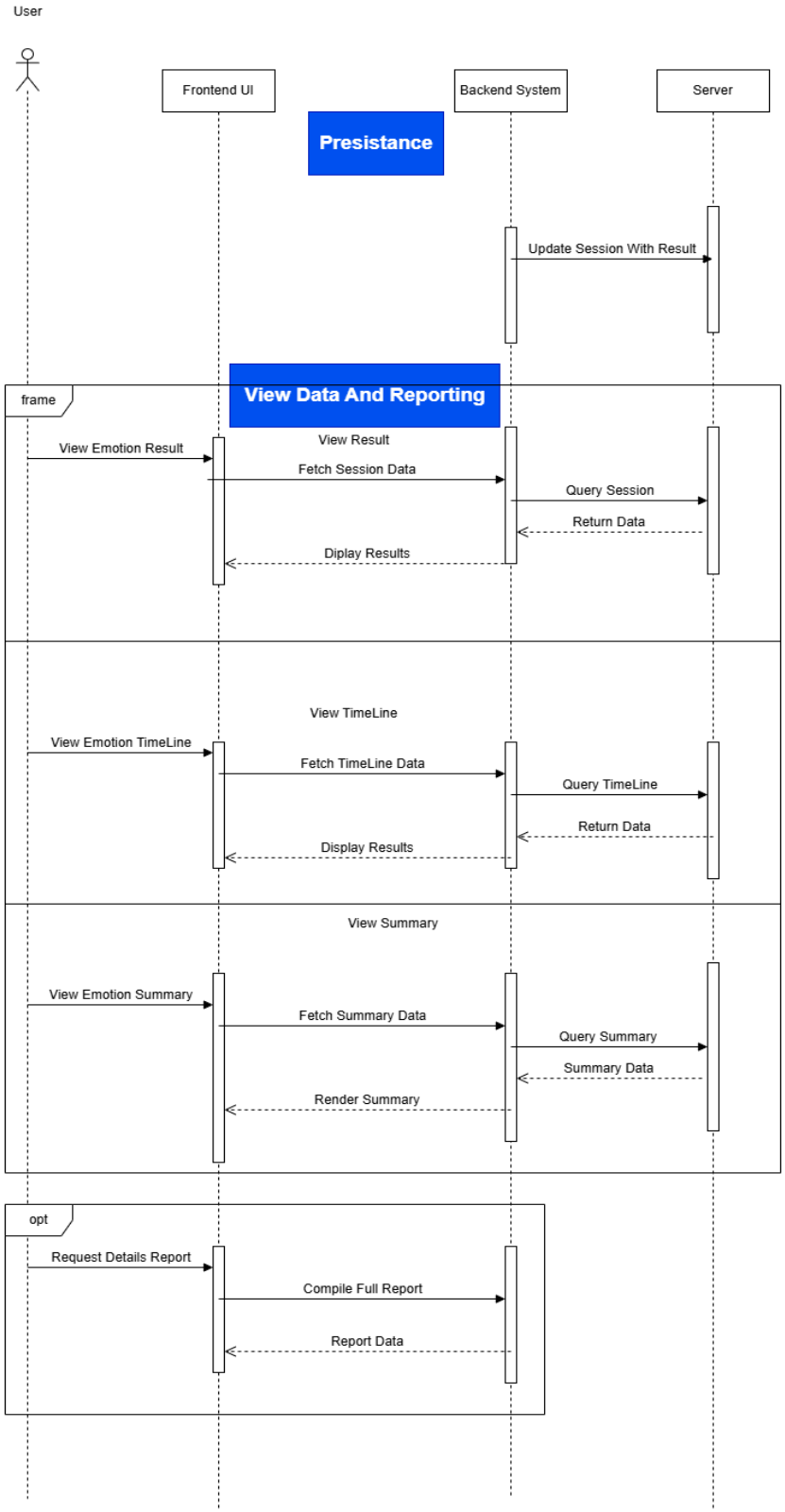
3.4.7.1. Authentication Sequence



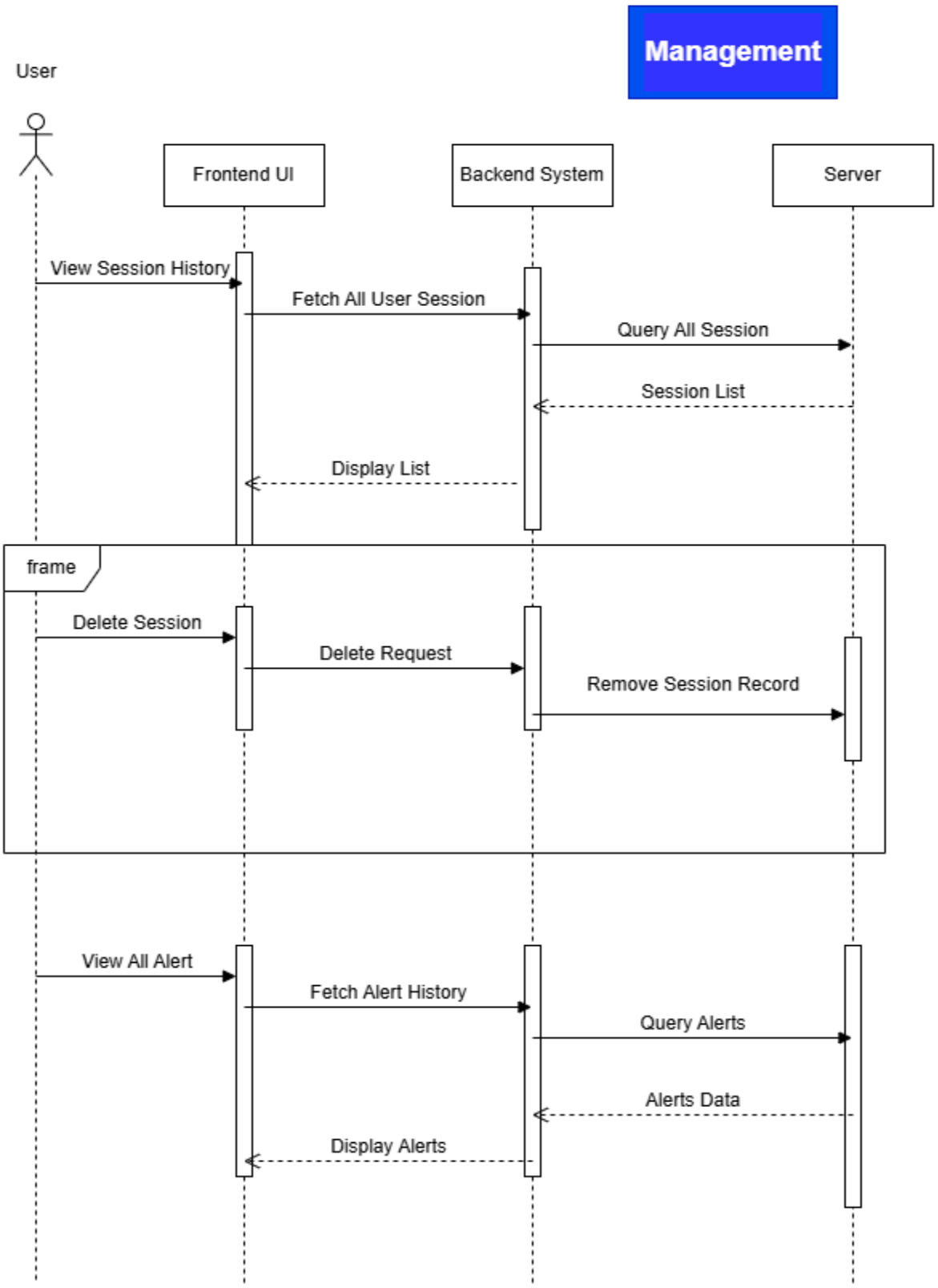
3.4.7.2. Upload Data Sequence



3.4.7.3. View Data Sequence



3.4.7.4. Management Sequence



Chapter 4: Artificial Intelligence Models

4.1 Introduction to AI Models Used

The Emotion Detection with Timeline Analysis system relies on artificial intelligence models to automatically identify and interpret human emotional states from multiple types of data. Unlike traditional emotion recognition systems that produce a single static emotion label, the proposed system is designed to capture the dynamic nature of emotions and represent them as a temporal sequence that evolves throughout an interaction or session.

To achieve this goal, the system adopts a model-driven yet system-oriented approach, where each input modality—text, audio, image, and video—is analyzed using a specialized AI model optimized for that data type. These models operate independently within modular AI microservices and generate structured emotion predictions that can later be aligned and visualized on a unified emotion timeline.

Rather than training deep learning models from scratch, the project utilizes pre-trained transformer-based and deep learning models sourced from reliable research and industry-standard repositories such as Hugging Face. This design choice ensures high accuracy, robustness, and practical feasibility within the constraints of an academic graduation project. By leveraging pre-trained models, the system focuses on emotion analysis logic, temporal interpretation, fusion strategies, and visualization, which represent the core innovation of the project.

Each AI model produces probabilistic emotion outputs instead of simple categorical labels. These probability distributions allow the system to:

- Measure emotion intensity and confidence
- Detect subtle emotional transitions
- Support aggregation and comparison across time
- Enable higher-level analytics such as dominant emotion detection and emotional trend analysis

The AI models are deployed as independent Python-based microservices, exposed through RESTful APIs. This architecture promotes scalability, maintainability, and flexibility, allowing individual models to be updated or replaced without affecting the overall system. The backend layer orchestrates communication between these AI services, stores analysis results, and prepares data for timeline visualization on the frontend interface.

Summary

In summary, the AI models used in this project serve as the analytical foundation of the

Emotion Detection with Timeline Analysis system. Their role extends beyond simple classification, enabling a continuous, interpretable, and multimodal understanding of emotional behavior over time. This approach transforms emotion detection from a static prediction task into a dynamic and meaningful analytical process.

4.2 Emotion Classification Framework

A unified emotion classification framework is essential for ensuring consistency and interpretability across the different AI models used in the Emotion Detection with Timeline Analysis system. Since the system analyzes emotions from multiple modalities—text, audio, image, and video—it is necessary to adopt a common emotional representation that allows results from different sources to be compared, aggregated, and visualized on a shared timeline.

The proposed system follows a discrete emotion classification approach, where each input segment is mapped to a predefined set of basic emotional states. This approach was chosen over dimensional models (such as valence–arousal) because discrete emotions are easier to interpret, visualize, and communicate to end users, particularly in dashboard and timeline-based interfaces.

4.2.1 Core Emotion Labels

Across all AI models, the system adopts the following seven core emotion labels:

- Anger
- Disgust
- Fear
- Joy
- Sadness
- Surprise
- Neutral

These emotions are widely used in emotion recognition research and are supported by the pre-trained models integrated into the system. Using a fixed and shared label set ensures that emotion outputs from different modalities remain compatible and can be processed using the same aggregation and visualization logic.

4.2.2 Emotion Category Mapping

To support higher-level analysis and alert generation, each emotion label is mapped into one of three broader emotional categories:

- **Positive:** Joy, Surprise
- **Negative:** Anger, Disgust, Fear, Sadness

- **Neutral:** Neutral

This categorization allows the system to compute meaningful emotional summaries, such as positive–negative balance, dominant emotional polarity, and long-term emotional trends. It also enables the detection of recurring negative emotional patterns, which are later used by the alerting mechanism.

4.2.3 Probabilistic Emotion Representation

Instead of returning a single emotion label, all AI models in the system produce probabilistic emotion outputs. Each prediction consists of confidence values associated with every supported emotion class. The dominant emotion is determined by the highest confidence score, while the remaining probabilities provide insight into secondary or mixed emotional states.

This probabilistic representation is critical for:

- Measuring emotional intensity
- Handling ambiguous or overlapping emotions
- Smoothing emotional fluctuations over time
- Supporting fusion between multiple analysis levels or modalities

4.2.4 Temporal Interpretation of Emotions

A central concept of the classification framework is the temporal interpretation of emotions. Emotions are not treated as isolated predictions but as a sequence of emotional states ordered over time.

Depending on the modality:

- Text is interpreted as a sequence of sentences
- Audio is segmented into time-based windows
- Video is analyzed frame by frame
- Images represent single time points

Each segment corresponds to a position on the emotion timeline, allowing the system to observe emotional transitions, stability, or sudden changes throughout a session. This design directly supports the project's core objective of transforming static emotion detection into timeline-based emotional analysis.

4.2.5 Framework Consistency Across Modalities

By enforcing a unified emotion classification framework, the system ensures that:

- All AI models follow the same emotional vocabulary
- Emotion outputs can be stored using a common database schema

- Timeline visualization logic remains consistent
- Multimodal comparison and fusion become feasible

This framework acts as a bridge between the AI models and the system-level features such as dashboards, historical tracking, and alert generation.

4.2.6 Summary

The emotion classification framework provides the structural foundation upon which all AI models in the system operate. By defining a consistent set of emotion labels, categories, probabilistic outputs, and temporal interpretation rules, the framework enables reliable emotion comparison, aggregation, and visualization across time and modalities. This unified approach is essential for achieving accurate and interpretable timeline-based emotion analysis in the proposed system.

4.3 Text-Based Emotion Detection Model

4.3.1 Model Overview

The text-based emotion detection model is one of the core analytical components of the Emotion Detection with Timeline Analysis system. Its primary purpose is to identify and interpret emotional states expressed through written language and to support the system's central objective of tracking how emotions evolve over time.

Textual input often represents complex emotional expression. A single paragraph may contain calm descriptions, moments of sadness, brief optimism, or emotional tension. Treating such input as a single emotional unit would result in a loss of valuable emotional information. For this reason, the text-based model is designed not only to classify emotions, but also to preserve emotional variation within the input.

The model operates by analyzing text at multiple levels, allowing it to capture both localized emotional expressions and the overall emotional tone of the input. This multi-level design ensures that emotional transitions within the text are detected and can later be visualized as part of an emotion timeline.

Within the system architecture, the text emotion model functions as an independent AI microservice. It receives raw text input, processes it using deep learning techniques, and returns structured emotion data. This data is later stored, aggregated, and visualized by the backend and frontend components of the system. As such, the text model serves as a foundational building block for timeline-based emotional analysis.

4.3.2 Model Selection and Justification

The text-based emotion detection component uses a transformer-based deep learning

model obtained from Hugging Face, specifically the j-hartmann/emotion-english-distilroberta-base model. This model is based on the DistilRoBERTa architecture and is fine-tuned for multi-class emotion classification.

The selection of this model was driven by both technical and practical considerations. Transformer-based models are widely recognized for their ability to capture contextual meaning in natural language. Unlike traditional machine learning approaches or lexicon-based sentiment analysis methods, transformer models consider the relationship between words within their surrounding context, allowing them to better interpret emotionally nuanced sentences.

DistilRoBERTa was chosen in particular due to its balance between performance and efficiency. While larger transformer models may offer marginally higher accuracy, they also introduce higher computational costs. DistilRoBERTa provides strong emotion classification performance while remaining suitable for real-time inference in an API-based system, which is essential for a responsive web application.

Another important justification for this choice is the decision to rely on pre-trained models rather than training custom models from scratch. Training emotion detection models requires large, diverse, and carefully annotated datasets, as well as significant computational resources. By leveraging a well-established pre-trained model, the project ensures reliable emotion recognition performance while allowing development efforts to focus on system-level innovation, such as emotion timeline generation, multimodal integration, and visualization.

This approach aligns with the academic scope of the project, where the contribution lies in AI system design and integration, rather than low-level model training.

4.3.3 Emotion Labels and Categories

The text emotion detection model classifies input text into a predefined set of seven discrete emotion labels:

- Anger
- Disgust
- Fear
- Joy
- Sadness
- Surprise
- Neutral

These emotions were selected because they are widely used in emotion recognition research and are directly supported by the underlying pre-trained model. Using a fixed and well-

defined label set ensures consistency across different analyses and simplifies downstream processing.

To facilitate higher-level interpretation and system features such as alert generation and emotional trend analysis, the system groups these emotions into three broader categories:

- **Positive emotions:** Joy, Surprise
- **Negative emotions:** Anger, Disgust, Fear, Sadness
- **Neutral emotions:** Neutral

This categorization enables the system to compute meaningful emotional summaries, such as the ratio of positive to negative emotions, the dominant emotional polarity of a session, and the detection of repeated negative emotional patterns over time. These summaries play a key role in supporting long-term emotional tracking and user awareness.

By maintaining a consistent emotion label and category structure across all modalities, the system ensures that text-based emotion results can be directly compared and combined with results from audio, image, and video analysis.

4.3.4 Text Preprocessing and Sentence Segmentation

Before emotion classification is performed, the input text undergoes a preprocessing stage focused on sentence segmentation. The text is divided into individual sentences using punctuation-based splitting, treating each sentence as a separate analytical unit.

Sentence segmentation is a critical design decision for timeline-based emotion analysis. Long textual inputs often contain multiple emotional shifts, and analyzing the text as a single block would obscure these transitions. By breaking the text into sentences, the system preserves localized emotional expressions and allows emotional changes to be detected more accurately.

Each sentence retains its original order within the text, which later enables the system to interpret sentence sequence as a temporal progression. This approach allows textual data to be aligned conceptually with other modalities, such as audio segments or video frames, even though text does not contain explicit timestamps.

Summary

The segmentation process ensures that emotionally significant sentences are not diluted by surrounding neutral content. As a result, the model can produce a more detailed and interpretable emotional representation, forming the basis for sentence-level analysis and emotion timeline construction.

4.3.5 Sentence-Level Emotion Analysis

After preprocessing and sentence segmentation, each sentence is analyzed independently by the text-based emotion detection model. This sentence-level analysis is a central component of the system's timeline-based design, as it enables the detection of emotional variation within a single textual input.

For every sentence, the transformer-based model generates a probability distribution across all supported emotion labels. These probabilities represent the model's confidence that a given sentence expresses each specific emotion. The emotion with the highest confidence score is identified as the dominant emotion for that sentence.

By processing sentences individually, the system avoids the limitations of traditional text emotion analysis approaches that rely on a single global prediction. This method allows subtle emotional shifts to be captured, such as a transition from neutral description to sadness or from tension to relief within the same text.

The ordered sequence of sentence-level predictions forms the foundation of the text emotion timeline. Each sentence represents a discrete emotional point, and together they provide a detailed emotional progression that reflects how the emotional state evolves throughout the text.

4.3.6 Emotional Intensity Weighting Mechanism

While sentence-level analysis captures emotional variation, not all sentences contribute equally to the emotional meaning of a text. Some sentences carry strong emotional significance, while others serve a descriptive or neutral role. To address this, the system incorporates a custom emotional intensity weighting mechanism.

This mechanism adjusts the influence of each sentence based on the presence of emotionally expressive language. Sentences containing strong emotional indicators—such as words associated with intense fear, sadness, anger, or disgust—are assigned higher weights. Conversely, sentences with mild or neutral expressions are assigned lower weights.

The weighting process ensures that emotionally powerful sentences have a greater impact on the final emotion calculation. This prevents emotionally significant moments from being overshadowed by longer sections of neutral content and improves the realism of the emotion analysis.

By incorporating intensity weighting, the system goes beyond simple probability aggregation and introduces an additional layer of semantic interpretation that reflects emotional strength, not just emotional type.

4.3.7 Full-Text Emotion Analysis

In addition to sentence-level analysis, the system performs a full-text emotion analysis by processing the entire text input as a single unit. This analysis captures the overall emotional context and provides a global emotional perspective.

The full-text analysis serves several important purposes. First, it reduces noise that may arise from sentence-level fragmentation, particularly in cases where individual sentences are short or ambiguous. Second, it ensures that the final emotion result remains stable and representative of the input as a whole.

The model generates a probability distribution for the full text in the same manner as it does for individual sentences. This global prediction reflects the dominant emotional tone that emerges when the text is considered in its entirety.

By combining local and global emotional information, the system achieves a more balanced and reliable emotional interpretation.

4.3.8 Emotion Fusion Strategy

To produce the final emotion output, the system applies a hybrid emotion fusion strategy that combines the results of sentence-level analysis with the full-text emotion analysis.

In this strategy, weighted sentence-level emotion probabilities are aggregated to represent localized emotional expression, while the full-text emotion probabilities contribute a stabilizing global influence. A predefined global weight is applied to the full-text results to ensure that they influence the final outcome without suppressing sentence-level variation.

This fusion approach balances sensitivity and stability. It preserves emotional transitions detected at the sentence level while preventing extreme fluctuations caused by isolated or ambiguous sentences.

The result of this fusion process is a comprehensive emotional profile that includes:

- A dominant final emotion
- Confidence scores for all emotion labels
- A structured emotional progression suitable for timeline visualization

This fusion strategy plays a key role in aligning the text-based emotion detection model with the system's broader objective of timeline-based emotional analysis.

4.3.9 Timeline Interpretation for Text

Although textual data does not contain explicit timestamps, the system interprets text as a logical temporal sequence based on sentence order. Each sentence represents a successive point in time, allowing emotional changes to be analyzed in a manner consistent with other modalities such as audio and video.

By treating sentence order as temporal progression, the system transforms static text into a form suitable for timeline-based analysis. Emotional transitions—such as shifts from neutrality to sadness or from tension to relief—can be observed and visualized along the timeline.

This interpretation ensures conceptual consistency across all emotion detection modalities. Text sentences, audio segments, and video frames are all represented as ordered emotional units, enabling unified visualization and comparison within the same framework. As a result, text-based emotion analysis integrates seamlessly into the system's global emotion timeline.

4.3.10 API and Microservice Integration

The text-based emotion detection model is deployed as an independent API-driven microservice implemented using FastAPI. This service exposes a RESTful endpoint that accepts textual input and returns structured emotion analysis results in JSON format.

The microservice architecture offers several advantages:

- Loose coupling between the AI model and backend logic
- Improved scalability and maintainability
- The ability to update or replace the model without affecting other system components

The backend system communicates with the text emotion API to store analysis results, generate emotion timelines, and perform higher-level processing such as emotional trend detection and alert generation. This modular design supports future expansion and integration of additional AI models.

4.3.11 Model Limitations

Despite its effectiveness, the text-based emotion detection model has inherent limitations. The model may struggle with detecting sarcasm, irony, or implicit emotional cues that rely heavily on cultural or contextual understanding. Additionally, variations in writing style, grammar, and vocabulary can influence prediction accuracy.

The model is also limited to processing English text and does not account for multilingual or cross-cultural emotional expression. Furthermore, as the system is designed for emotional awareness and analysis, it does not provide clinical or psychological diagnoses.

These limitations are acknowledged within the project and are partially mitigated through multimodal emotion detection, where textual analysis is complemented by audio and visual emotion recognition to provide a more comprehensive understanding of emotional behavior.

4.4 Audio-Based Emotion Detection Model

4.4.1 Model Overview

The audio-based emotion detection model is designed to analyze human speech and identify emotional states expressed through vocal signals. Speech conveys emotion not only through spoken words, but also through acoustic characteristics such as tone, intensity, rhythm, and energy. For this reason, audio emotion analysis plays a critical role in achieving a comprehensive understanding of emotional behavior within the proposed system.

Unlike traditional speech emotion recognition systems that rely solely on acoustic features, the proposed model adopts a multimodal speech analysis approach. It combines acoustic emotion detection with linguistic emotion analysis extracted from transcribed speech content. This design enables the system to capture both how something is said and what is being said, resulting in more accurate and robust emotion recognition.

The audio emotion detection component operates as an independent AI microservice. It processes uploaded audio files, segments them into time-based units, performs emotion analysis, and generates a timeline of emotional states. The output is structured to integrate seamlessly with the backend system and align with the global emotion timeline used across all modalities.

4.4.2 Audio Preprocessing and Segmentation

Before emotion classification, the audio input undergoes a preprocessing stage to prepare it for analysis. The audio file is first converted into a standardized format by resampling it to a fixed sampling rate and converting it to a mono signal. This ensures compatibility with the underlying deep learning models and consistency across different audio sources.

To support timeline-based emotion analysis, the audio signal is divided into fixed-length segments, each representing a short time window within the overall recording. In the proposed system, each segment corresponds to a one-second interval. This segmentation strategy allows emotional changes to be detected at a fine temporal resolution while maintaining computational efficiency.

Each audio segment retains its timestamp offset relative to the beginning of the recording. These timestamps are later used to construct an emotion timeline that reflects how emotional states evolve throughout the audio. By segmenting the audio in this manner, the system avoids producing a single static emotion label and instead generates a detailed emotional

progression across time.

Additionally, an energy-based weighting mechanism is applied to each segment using the root mean square (RMS) of the audio signal. Segments with higher energy levels, which often indicate stronger emotional expression, are assigned greater influence during aggregation. This weighting improves the sensitivity of the analysis to emotionally intense moments within the audio.

4.4.3 Feature Extraction

Feature extraction is performed using a deep learning-based approach rather than traditional handcrafted audio features. The system employs a HuBERT-based speech emotion recognition model obtained from Hugging Face, specifically fine-tuned for speech emotion recognition tasks.

HuBERT is a self-supervised speech representation model that learns high-level acoustic features directly from raw audio waveforms. By relying on learned representations instead of manually engineered features, the model captures complex patterns related to emotional expression, such as intonation, stress, and temporal dynamics.

The feature extraction process is handled internally by the model's feature extractor, which converts each audio segment into a representation suitable for emotion classification. These extracted features are then passed to the emotion classification head to generate emotion probability scores.

This deep feature extraction approach provides robustness to background noise and speaker variation and ensures that the audio-based emotion detection model remains consistent with modern best practices in speech emotion recognition research.

4.4.4 Emotion Classification Process

After feature extraction, each audio segment is passed through the speech emotion recognition model to perform emotion classification. The model outputs a probability distribution representing the likelihood of each supported emotional state for the given segment. These probabilities reflect the acoustic emotional cues present in the speech, such as tone, pitch variation, and vocal intensity.

Because the underlying speech emotion recognition model natively supports a limited set of emotion classes, the system applies a label-space normalization strategy to align audio emotion outputs with the unified seven-emotion framework used across the entire system. This ensures consistency between text-based, audio-based, and visual emotion detection results.

In parallel with acoustic emotion classification, the system performs linguistic emotion analysis by transcribing the speech content into text using OpenAI Whisper. The transcribed text is then processed by the text-based emotion detection model described in Section 4.3. This dual-path analysis allows the system to capture both vocal expression and semantic emotional meaning.

To produce a final emotion representation for audio input, the system applies a late fusion strategy that combines acoustic emotion probabilities with linguistic emotion probabilities. This decision-level fusion enhances robustness, particularly in cases where one modality may be ambiguous or noisy.

4.4.5 Timeline Generation for Audio

A key objective of the audio-based emotion detection model is to support timeline-based emotional analysis. Each segmented audio window corresponds to a specific timestamp offset within the recording, allowing emotional predictions to be ordered chronologically.

For every audio segment, the system records:

- The timestamp offset
- Emotion probability distribution
- Dominant emotion label
- Confidence score
- Segment-level intensity weight

These segment-level results are stored as a sequence, forming an emotion timeline that represents how emotional states evolve throughout the audio recording. This timeline enables visualization of emotional fluctuations, detection of emotional peaks, and identification of prolonged emotional patterns.

In addition to the detailed timeline, the system aggregates weighted segment-level predictions to compute overall emotion statistics for the entire audio input. These aggregated results are used for summary visualization, historical tracking, and alert generation.

By treating audio as a sequence of emotional segments rather than a single unit, the system preserves temporal emotional dynamics and maintains consistency with timeline-based analysis applied to other modalities.

4.4.6 Model Limitations

Despite its effectiveness, the audio-based emotion detection model has certain limitations. Acoustic emotion recognition can be influenced by background noise, recording quality, and speaker variability. Differences in speaking style, accent, and speech rate may also affect

prediction accuracy.

The reliance on automatic speech transcription introduces additional constraints. Errors in transcription may impact the quality of linguistic emotion analysis, particularly in noisy environments or for non-native speakers. Furthermore, the speech emotion recognition model supports a limited native emotion set, requiring heuristic mapping to the unified emotion framework.

The model is also language-dependent and primarily optimized for English speech. Emotional expressions that are culturally specific or expressed through non-verbal vocalizations may not be fully captured.

These limitations are acknowledged within the project scope and are partially mitigated through multimodal analysis, where audio-based emotion detection is complemented by text and visual emotion recognition to provide a more comprehensive emotional understanding.