كلية الحاسبات والذكاء الاصطناعي
FACULTY OF COMPUTERS & ARTIFICIAL INTELLIGENCE

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

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 1

2. Problem Definition 2

2.1. The Static Output Problem (One-Shot Classification) 2

2.2. Lack of Multimodal Integration 2

2.3. Absence of Timeline-Based Analysis 3

2.4. User Experience and Tracking Limitations 3

2.5. Technical Scalability and Interoperability Gap 3

3. Proposed Solution 4

3.1. System Architecture and Layering 4

3.2. Functional Workflow 5

3.3. Key Features and Innovations 5

3.4. Expected Benefits 6

4. Literature Review 7

4.1. Introduction 7

4.2. Existing Facial Emotion Detection Systems 7

4.3. Existing Speech Emotion Recognition Systems 8

4.4. Existing Text Emotion and Sentiment Analysis Platforms 8

4.5. Existing Multimodal Emotion Recognition Systems 8

4.6. Existing Timeline-Based and Longitudinal Emotion Systems 9

4.7. Timeline and Longitudinal Emotion Analysis Gaps 9

4.8. Combined Limitations of Existing Systems 10

4.9. Summary 11

5. Project Objective 11

Detailed Objectives 11

6. Scope of the Project 12

In-Scope Features 12

Out of Scope (Phase 1) 13

Summary 13

7. Scope Exclusions and Constraints 13

Scope Exclusions 13

Constraints 14

8. Project Methodology 14

1. Planning Phase....14
2. Analysis Phase....15
3. Design Phase....15
4. Implementation Phase....15
5. Testing Phase....15
Deployment and Maintenance....16
Chapter 2: Project Management....17
2.1. Project Organization....17
Team Structure....17
Communication Workflow....17
2.2. Risk Management....17
Potential Risks and Mitigation Strategies....18
Summary....18
2.3. PROJECT COMMUNICATION PLAN....19
2.4. WORK BREAKDOWN STRUCTURE (WBS)....20
2.5. TIME MANAGEMENT....22
2.5.1. PERT Equation....22
2.5.2. Network diagram....23
2.5.3. Critical path....25
2.5.4. Gantt Chart....25

Chapter 1: Introduction & Background

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.

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.

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.

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.

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.

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.

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.

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

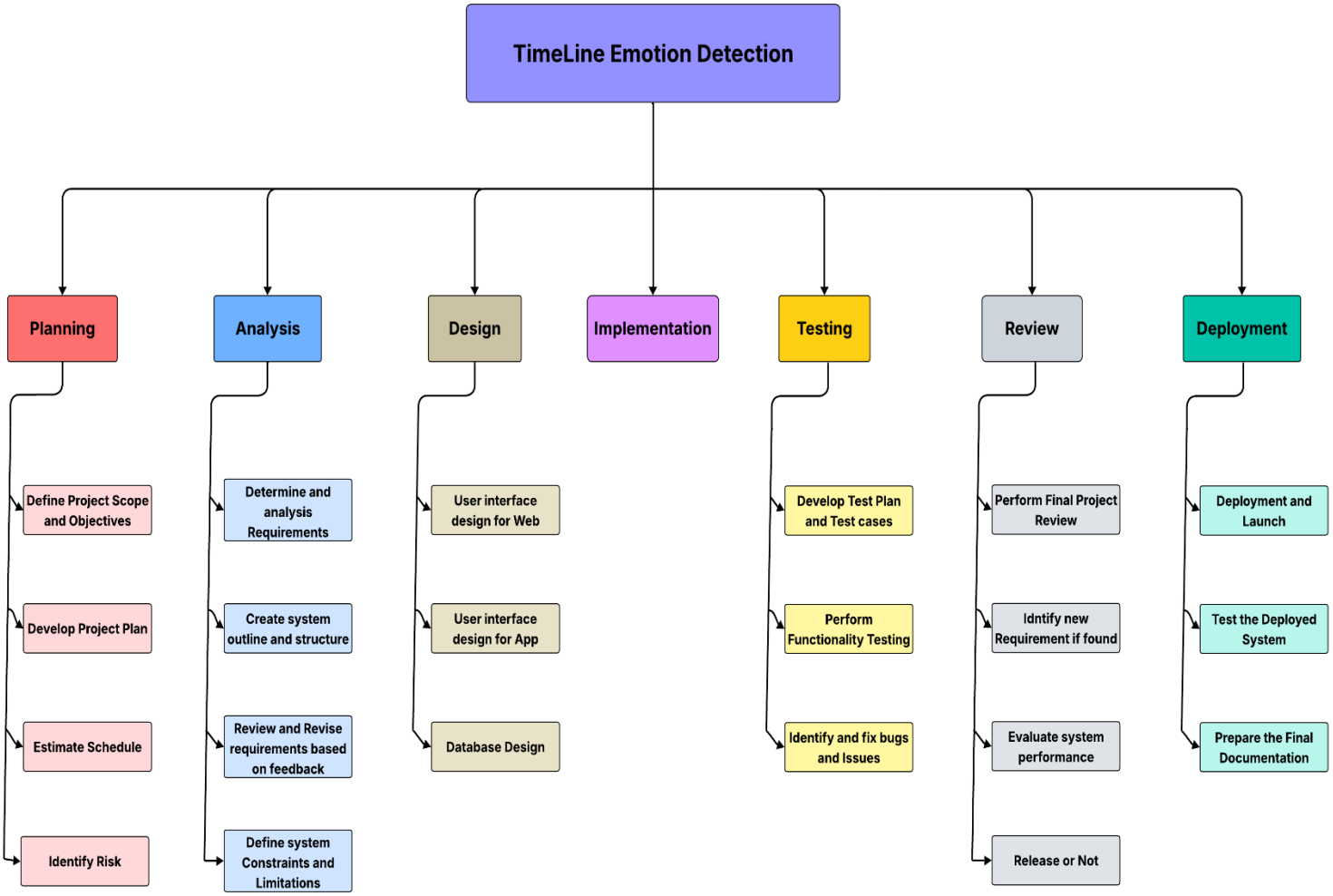
2.4. WORK BREAKDOWN STRUCTURE (WBS)

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.

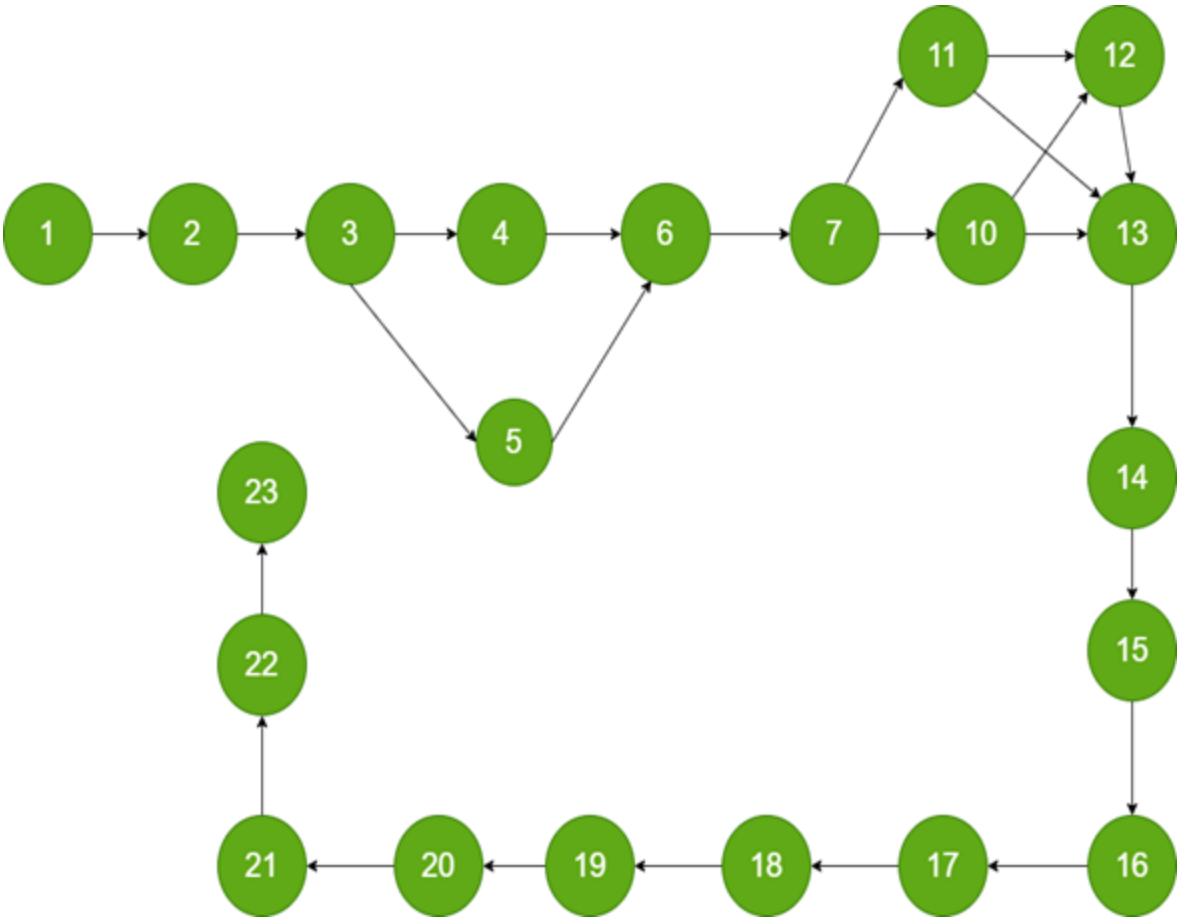
Note that: the time on the table is in Days.

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

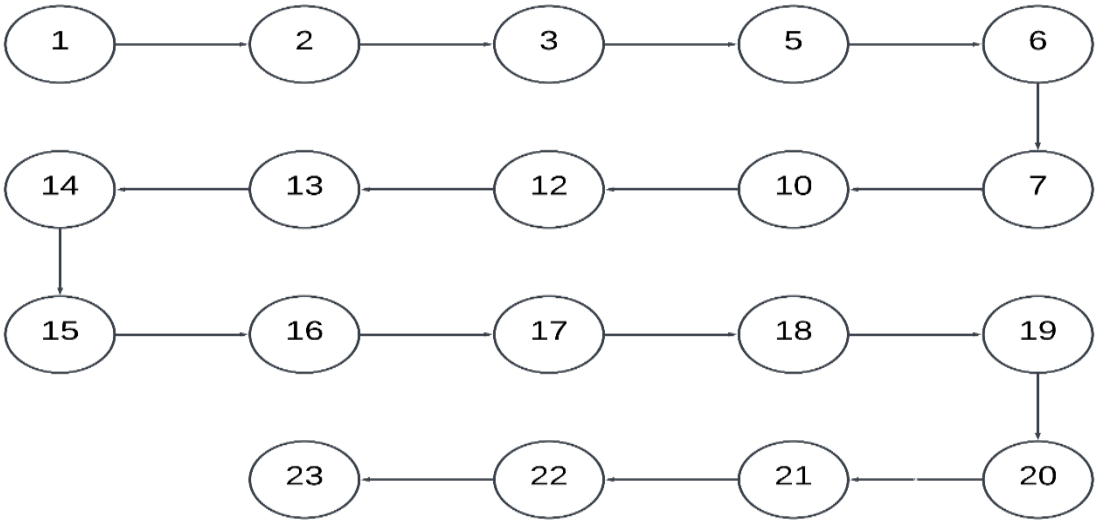
2.5.2. Network diagram

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