

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

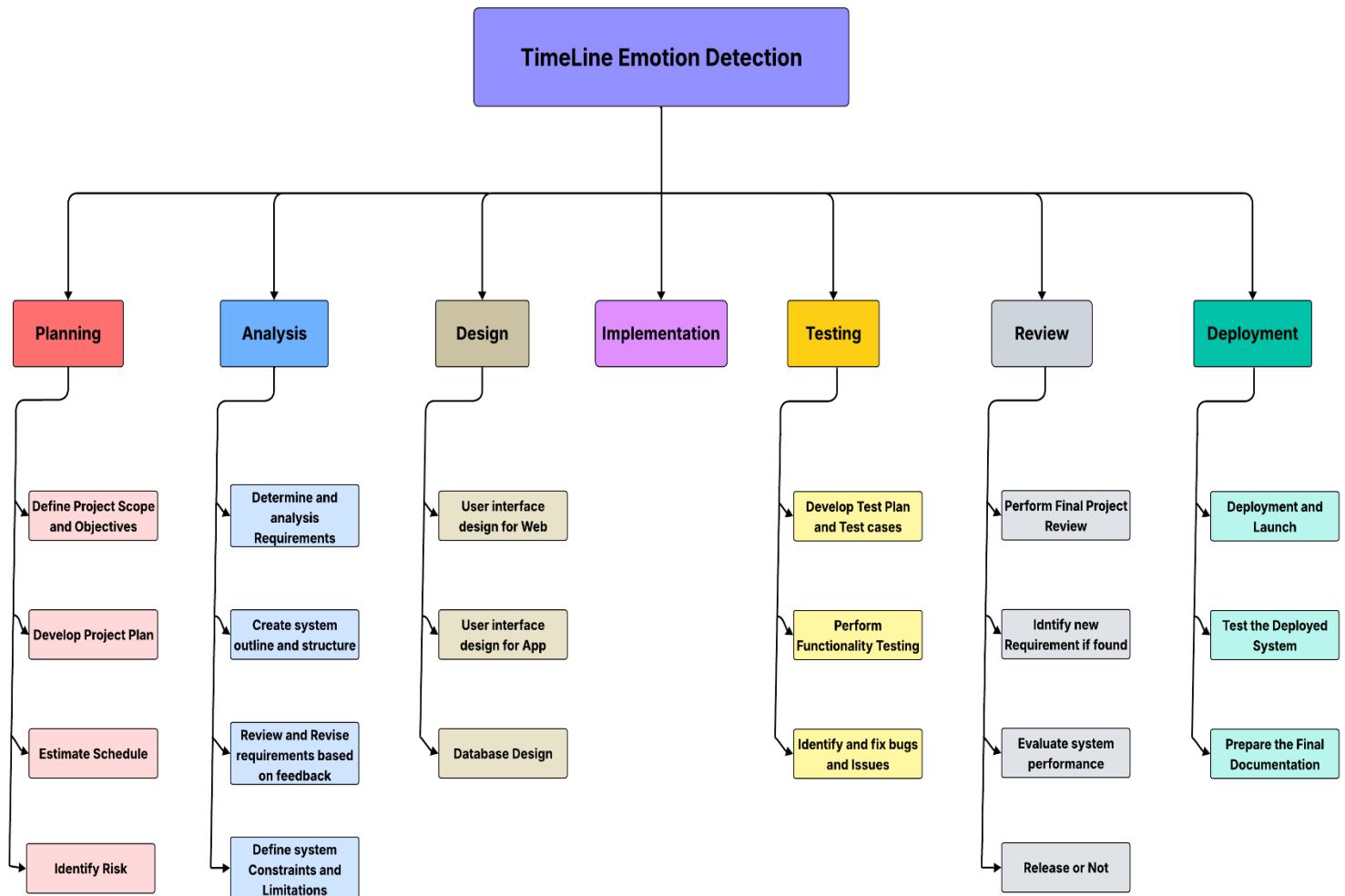
2.4. WORK BREAKDOWN STRUCTURE (WBS)

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.

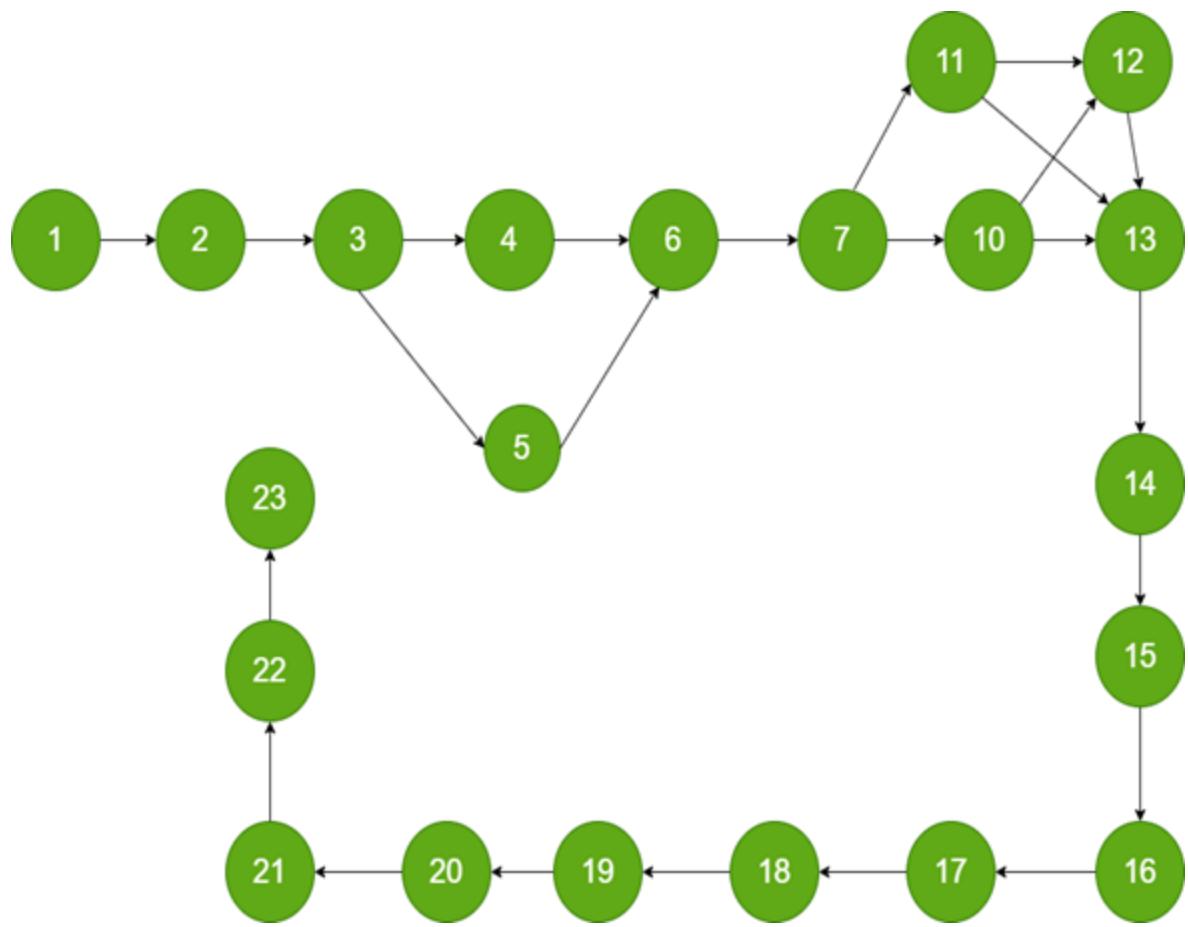
Note that: the time on the table is in Days.

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

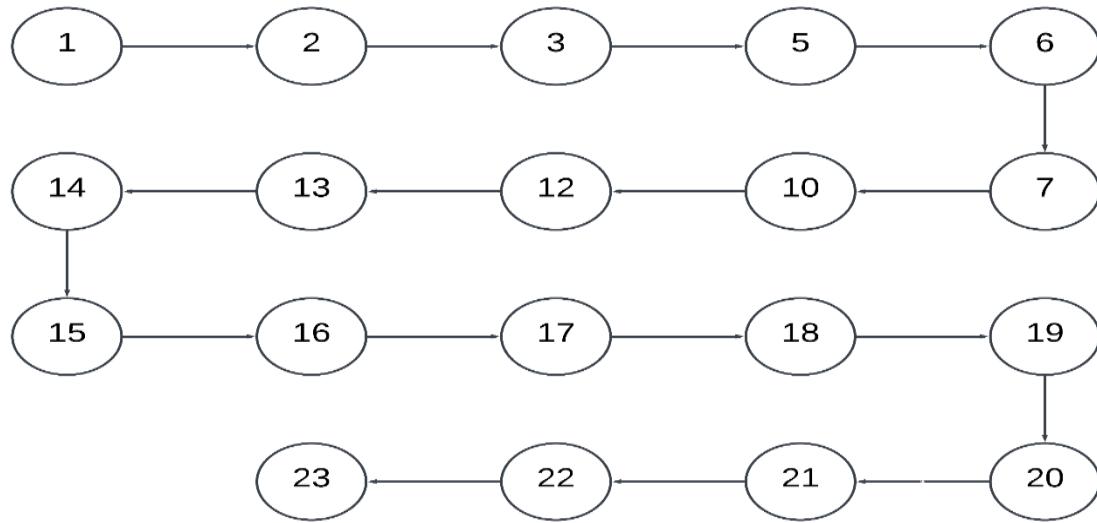
2.5.2. Network diagram

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



2.5.3. Critical path



2.5.4. Gantt Chart

Project Gantt Chart

