Software Requirements Specification

For

Comment ML – YouTube Sentimental & Trend Analyzer

09-October-2025

Prepared by

Specialization	SAP ID	Name
MCA-Aiml	590016150	Suraj Kumar
MCA-Aiml	590016745	Lucky Patel
MCA-Aiml	590016229	Siddharth Kumar
MCA-Aiml	590017783	Saloni Chaudhary

UPES
UNIVERSITY OF THE FUTURE

AI Cluster
School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

Table of Contents

Revision History

Date	Change	Reason for Changes	Mentor Signature
05 October 2025	Data Cleaning Process	Data are Uncleaned	[illegible] 10/10/25
09 October 2025	Data Cleaning Process	Data are Uncleaned	[illegible] 10/10/25
09 October 2025	Exploratory Data Analysis	To find Hidden pattern	[illegible] 10/10/25

1. INTRODUCTION

1.1 Purpose of the Project

The purpose of **Comment ML – YouTube Sentimental & Trend Analyzer** is to automate the process of analyzing YouTube comments using Artificial Intelligence and Natural LanguageProcessing(NLP).
Millions of comments are posted daily on YouTube videos, providing creators with a direct line to their audience. However, the overwhelming volume and unstructured nature of these comments make it difficult to manually interpret sentiment or detect emerging trends.
This project aims to build an intelligent and interactive system that processes raw comments, classifies them as **positive**, **negative**, or **neutral**, detects trending keywords, and visualizes engagement patterns.
The proposed solution enables **YouTube creators, influencers, marketers, and researchers** to gain meaningful insights from comment data, improving content strategy and audience understanding.
By integrating machine learning and real-time visualization, the system bridges the gap between unstructured feedback and actionable insights. The project is also designed as a **Chrome Extension** with a backend built on **Flask/FastAPI**, allowing users to analyze sentiment directly from YouTube video pages.

1.2 Target Beneficiary

The system primarily benefits:

- **YouTube Creators & Influencers:** They can analyze audience responses and adjust future content accordingly.
- **Digital Marketing Teams:** To track brand reputation, campaign reactions, and audience engagement patterns.
- **Viewers & Audiences:** Their feedback gains greater visibility and is effectively represented.
- **Research Scholars & Analysts:** Provides sentiment data for studying online behavior and social trends.
- **Organizations & Media Agencies:** Enables large-scale comment analytics for trend tracking and competitive insights.

1.3 Project Scope

The project covers the **end-to-end pipeline** of comment analysis:

1. **Data Collection:** Fetching YouTube comments using the YouTube Data API or web scraping.
2. **Data Cleaning & Preprocessing:** Removing noise such as stopwords, punctuation, emojis, and duplicates; lemmatization for text normalization.
3. **Sentiment Classification:** Using ML/NLP algorithms like Logistic Regression, Random Forest, or BERT-based transformers.
4. **Trend & Keyword Detection:** Identifying frequently discussed terms or topics using word frequency and topic modeling.

5. **Visualization:** Generating charts, sentiment graphs, and word clouds to summarize insights.
6. **User Interaction:** Providing a Chrome Extension for real-time analytics with an intuitive UI.

1.4 Deliverables include:

- Cleaned dataset
- Trained sentiment analysis model
- Visual analytics dashboard
- Chrome Extension for user access

1.5 References

- https://raw.githubusercontent.com/Himanshu-1703/reddit-sentiment-analysis/main/data/reddit.csv
- YouTube Data API documentation
- NLTK and SpaCy NLP libraries
- Scikit-learn, TensorFlow, and Optuna framework documentation
- Research papers on sentiment analysis and trend detection (2020–2024)

2. PROJECT DESCRIPTION

2.1 Reference Algorithm

The reference methodology for sentiment analysis is based on a **supervised machine learning approach** enhanced with modern NLP techniques.
The pipeline includes:
TF-IDF Vectorization: To convert textual data into numerical form.
Feature Extraction: N-gram features and token frequency distributions.
Model Training: Logistic Regression, Naïve Bayes, or Transformer models such as BERT fine-tuned on labelled sentiment datasets.
Evaluation Metrics: Accuracy, Precision, Recall, and F1-score.
For trend detection, frequency-based analysis and topic modeling (using Latent Dirichlet Allocation) are employed to highlight recurring themes and discussions.
This hybrid approach ensures that the model not only captures simple lexical patterns but also context-sensitive sentiment.

2.2 Data / Data Structure

The dataset used in this project contains **37,249 labelled comments**, derived from Reddit and YouTube-stylediscussions.
Data Fields:

- clean_comment – pre-processed comment text
- category – sentiment label: *1 = positive, 0 = neutral, -1 = negative*

Preprocessing Workflow:

1. Remove null and duplicate entries.
2. Normalize text by converting to lowercase.
3. Remove URLs, emojis, and newline characters.
4. Tokenize and lemmatize words.

5. Remove stopwords and special symbols.
6. Create new features such as word count, character count, and stopword frequency.

Statistical Summary:

- Total records: 37,249
- Average word count: 29.6
- Positive: 42.8%, Neutral: 34.7%, Negative: 22.4%

Strengths	**Weaknesses**
Uses modern NLP and ML models	Limited initial dataset size
Real-time Chrome Extension	Handles English comments only
Visual and user-friendly dashboard	May misinterpret sarcasm or humor

2.3 Project Features

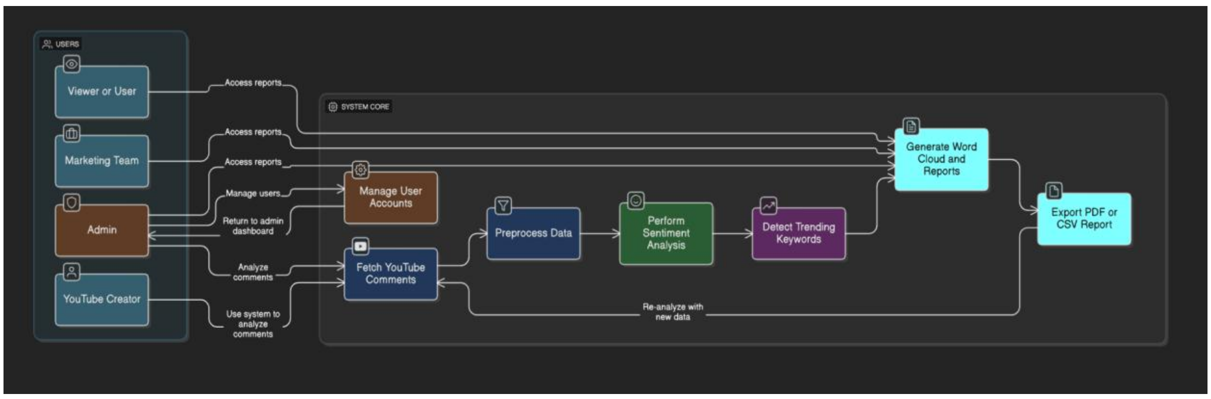
- Sentiment classification (positive, neutral, negative)
- Trend detection and keyword analysis
- Spam/troll filtering using ML heuristics
- Word cloud and sentiment distribution visualization
- PDF/CSV report generation
- Chrome Extension integration
- Real-time comment analysis with API connectivity

User Class	**Description**
YouTube Creators	View insights from their videos
Marketing Analysts	Measure public sentiment for brands
Viewers	Observe trends and express opinions
Researchers	Study online behavioral data
Administrators	Manage models and system updates

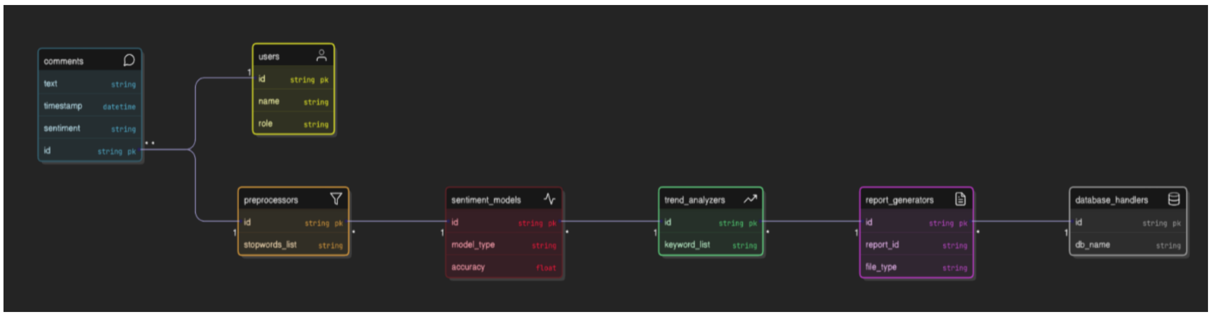
2.4 Design and Implementation Constraints

- Requires YouTube API access and stable internet.
- Minimum hardware: 8 GB RAM, dual-core processor.
- Environment: Python 3.10+, Flask/FastAPI, Docker.
- Only English language supported in the first phase.
- Must comply with privacy and YouTube data regulations.
- Chrome extension limited by Google’s manifest permissions.

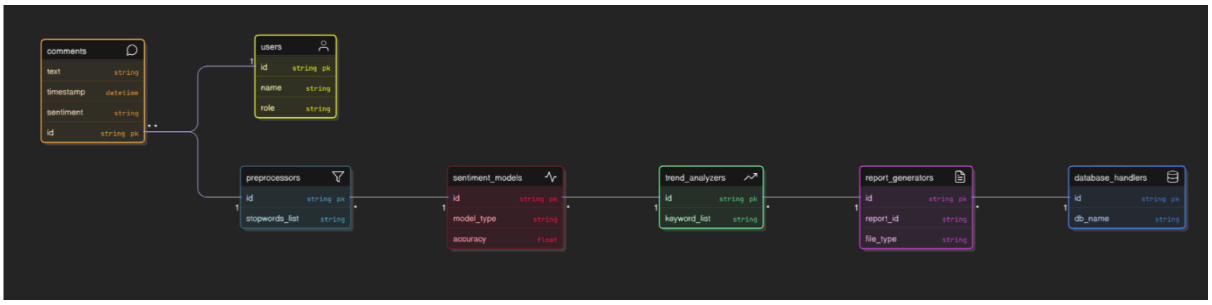
2.5 Design Diagrams



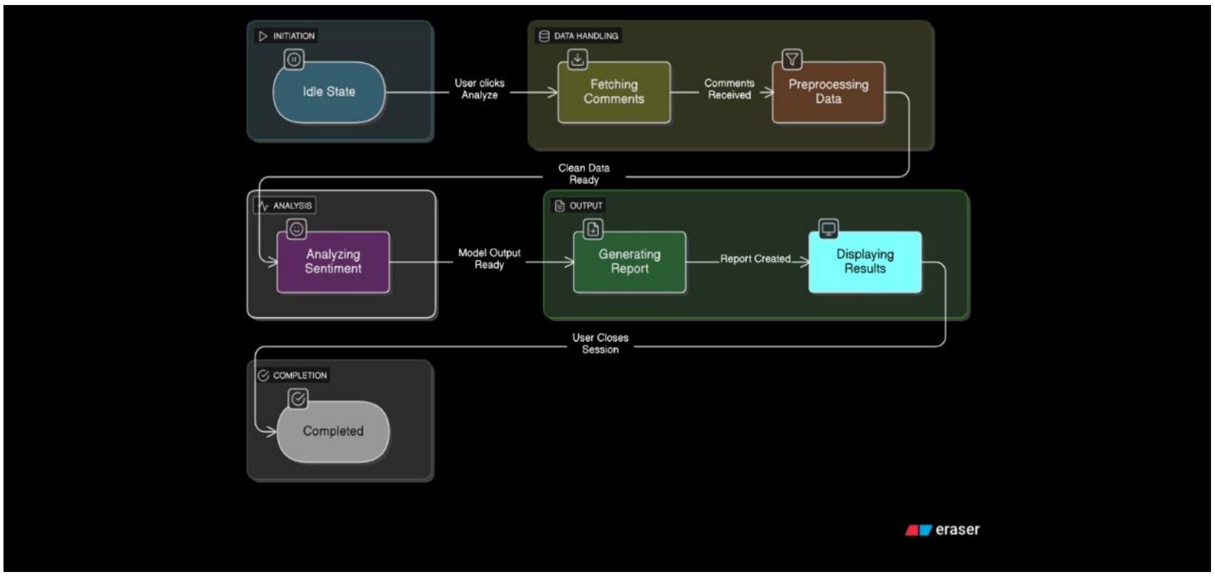
- **Use Case Diagram**



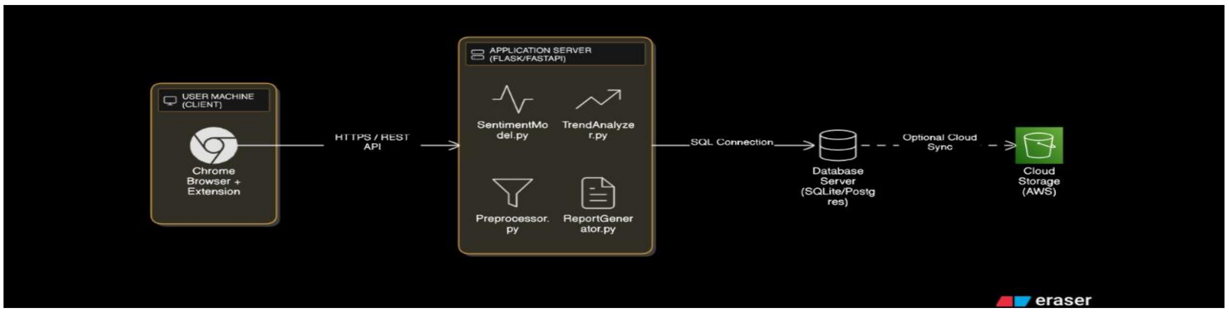
- **Class Diagram**



- **Data Flow Diagram (Level 0)**



- **State Diagram**



Deployment Diagram

2.6 Assumptions and Dependencies

- Assumes users have valid API access keys.
- Depends on continuous availability of YouTube Data API.
- Assumes moderate dataset size manageable within system memory.
- External dependencies include Flask, Seaborn, Matplotlib, and wordcloud libraries.
- Relies on pretrained language models like NLTK or BERT.

3. SYSTEM REQUIREMENTS

3.1 User Interface

The system offers a **Chrome Extension** with an integrated dashboard where users can:

- View comment sentiment distribution graphs
- Analyze trending keywords in real time
- View positive, negative, and neutral comment ratios
- Export reports in CSV or PDF format
- Access visualization modules such as bar charts, density plots, and word clouds

Backend results are fetched through API calls and displayed in a clean, responsive layout.

3.2 Software Interface

- **Frontend:** Chrome Extension (HTML, CSS, JavaScript)
- **Backend:** Flask/FastAPI REST API (Python)
- **Frameworks:** Scikit-learn, NLTK, SpaCy, TensorFlow
- **Visualization:** Matplotlib, Plotly, Seaborn
- **Communication:** JSON-based REST endpoints

3.3 Database Interface

- Database: SQLite or PostgreSQL
- Stores comment data, sentiment scores, and trend logs
- Schema includes:
    - user_data (user_id, email, role)
    - comment_data (comment_id, text, category)
    - analytics (keyword, frequency, date)
- Supports CRUD operations and data export

3.4 Protocols

- **Authentication:** OAuth 2.0 (for YouTube API)
- **Data Transfer:** HTTPS protocol for secure requests
- **Encryption:** SHA-256 encryption for credentials
- **API Communication:** RESTful JSON
- **Synchronization:** Real-time comment fetching

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Performance Requirements

- Response time per request: ≤ 3 seconds for 1000 comments
- Model accuracy: $\geq 85\%$
- System scalability up to 50,000 comments per session
- Real-time dashboard updates with minimal lag
- Support for concurrent Chrome Extension users

4.2 Security Requirements

- Data anonymization during preprocessing
- Encrypted API keys stored in environment variables
- Access control through user authentication
- HTTPS and token-based verification
- Compliance with YouTube and GDPR privacy policies

4.3 Software Quality Attributes

Attribute	Description
Adaptability	Easily adaptable to other platforms like Twitter or Instagram
Availability	99% uptime on AWS cloud
Maintainability	Modular code for quick updates
Portability	Deployed via Docker containers
Reliability	Tested across multiple datasets
Reusability	Reusable NLP modules
Usability	Simple Chrome dashboard interface
Testability	Unit and integration testing applied
Flexibility	Easily extendable model pipeline

5. OTHER REQUIREMENTS

- Must support multilingual analysis in future updates.
- Should include automatic report generation and email alerts.
- Optional integration with Google Sheets for data export.
- Maintain compatibility with both Windows and Linux platforms.

Appendix A: Glossary

Term	Meaning
NLP	Natural Language Processing
ML	Machine Learning
API	Application Programming Interface
TF-IDF	Term Frequency–Inverse Document Frequency
LDA	Latent Dirichlet Allocation
CI/CD	Continuous Integration / Continuous Deployment
EDA	Exploratory Data Analysis
BERT	Bidirectional Encoder Representations from Transformers

Appendix B: Analysis Model

- Sentiment category distribution visualization
- Word count and comment length analysis
- Frequency of stop words and special characters
- Top n-grams (bigrams/trigrams) analysis
- Word clouds for positive, negative, and neutral comments
- Density plots and boxplots of sentiment features

Appendix C: Issues List

Issue	**Status**	**Remarks**
Handling multilingual comments	Pending	Add translation pipeline
Sarcasm & emoji interpretation	Under review	Explore transformer models
Chrome Extension testing	Completed	API integrated successfully
Model optimization with Optuna	In progress	To improve accuracy
Real-time trend graph refresh	Pending	To be integrated with websocket