Software Requirements Specification

for

Inappropriate Comments Scanner

Prepared by

Adithya Singh	217Z1A6702	adithyasingh95@gmail.com
C. Harini Reddy	217Z1A6713	chereddyharini@gmail.com
N. Hema Narendra Reddy	217Z1A6743	reddynarendra408@gmail.com

Instructor: S. Ravi Teja

Contents

1 Introduction

1.1 Document Purpose

This document describes the software requirements for the Inappropriate Comments Scanner. The purpose of this system is to detect and filter inappropriate, offensive, or harmful content using Natural Language Processing (NLP) and Machine Learning (ML) algorithms. The system ensures that online communities maintain a respectful and safe environment by flagging inappropriate comments in real-time.

1.2 Product Scope

The Inappropriate Comments Scanner is a content moderation tool that integrates with social media platforms, forums, and educational websites. The system will:

- Use NLP models to detect hate speech, abusive language, and discriminatory remarks.
- Provide customizable filters to allow admins to define rules for content moderation.
- Support multiple languages to enhance accessibility and usability.
- Enable real-time monitoring to moderate conversations effectively.
- Offer a user-friendly interface for administrators to review flagged content.

1.3 Intended Audience and Document Overview

This document is intended for:

- Developers implementing the system.
- Project managers overseeing the development.
- Security teams ensuring compliance with moderation policies.
- End-users (administrators) who will review flagged content.

This document provides:

- An overview of the system.
- Detailed functional and non-functional requirements.
- Technical constraints and dependencies.

1.4 Definitions, Acronyms, and Abbreviations

- **NLP**: Natural Language Processing
- **ML**: Machine Learning
- **AI**: Artificial Intelligence
- **Hate Speech**: Content that promotes violence or discrimination
- **Conent Moderation**: Process of filtering or flagging harmful comments

1.5 Document Conventions

- Font: Arial, size 11
- Sections follow IEEE SRS formatting
- Diagrams and figures included where necessary

1.6 References and Acknowledgments

- IEEE SRS Template
- Research papers on content moderation using NLP
- NLP libraries such as TensorFlow, Pytorch, and TextBlob

2 Overall Description

2.1 Product Overview

The Inappropriate Comments Scanner is a web-based or API-integrated system that identifies and filters inappropriate user-generated content. It leverages NLP-based text classification models to detect offensive language, making it a powerful tool for real-time moderation.

2.2 Product Functionality

- Accept user-generated text input from various platforms.
- Pre-process text (tokenization, stopword removal, etc.)
- Use ML-based classifications models to detect inappropriate content.
- Generate moderation alerts for flagged content.
- Allow administrators to review flagged comments before final actions.
- Support language filtering and customization of detection parameters.

2.3 Design and Implementation Constraints

- The system must use Python-based NLP libraries (TextBlob, Scikit-learn, TensorFlow).Requires GPU for efficient vector processing
- Needs a real-time processing pipeline for instant detection.

2.4 Assumptions and Dependencies

- The system assumes a stable internet connection for API integration.
- Relies on pre-trained NLP models for accurate text classification.
- Users provide meaningful textual data

3 Specific Requirements

3.1 External Interface Requirements

- User Interface: Web-based dashboard for monitoring flagged content.
- Hardware Interfaces: Requires cloud-based servers or local GPU machines for efficient processing
- Software Interfaces: Integrates with external NLP libraries and databases

3.2 Functional Requirements

- F1: The system shall allow real-time comment analysis using NLP models
- F2: The system shall support multiple languages for text moderation
- F3: The system shall classify comments as safe, warning, or inappropriate
- F4: The system shall provide customizable filtering rules
- F5: The system shall generate moderation logs for admins to review flagged content

…

4 Other Non-functional Requirements

4.1 Performance Requirements

- The system shall analyze comments within 2 seconds for real-time detection.
- The NLP model shall achieve at least 90% accuracy in classification.

4.2 Safety and Security Requirements

- All flagged comments shall be stored in an encrypted database.
- Only authorized users (admins) shall have access to moderation logs.

4.3 Software Quality Attributes

- Reliability: The system shall maintain high uptime and handle document uploads efficiently.
- Usability: The UI shall be intuitive and allow users to easily navigate clustered documents.
- Scalability: The system shall support large-scale document processing without performance degradation.

5 Other Requirements

- Database storage shall support NoSQL for flexibility in handling document embeddings.

- The system shall support REST API integration for third-party applications.