**Software Requirements Specification (SRS)**

**Emotion Recognition System**

**Prepared by:**

**Samraan Ahmed**

**Iqra Liaqat**

**Hijab Zahra**

**Sara Hussain**

**Submitted to:**

**Dr. Mumtaz Zahoor & Mr. Farhan Zafar**

**Department:**

**BS Computer Science**

**Semester:**

**4th**

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1. Overall Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

1. Specific Requirements

3.1 Functional Requirements

3.2 Non-Functional Requirements

3.3 External Interface Requirements

1. System Features
2. Other Non-Functional Requirements

5.1 Performance Requirements

5.2 Security Requirements

5.3 Quality Attributes

1. Appendix

6.1 Glossary

6.2 Additional Notes

**1. Introduction**

**1.1 Purpose**

This Software Requirements Specification (SRS) document defines the requirements for the Emotion Recognition System, a software application designed to analyze and predict emotions from text and audio inputs. The document serves as a guide for developers, stakeholders, and evaluators to understand the system’s functionality, scope, and technical specifications.

**1.2 Scope**

The Emotion Recognition System is an AI-powered application developed to detect emotions such as sadness, joy, love, anger, fear, and surprise from text inputs, and neutral, calm, happy, sad, angry, fearful, disgust, and surprised from audio inputs. Built using Python and a custom Streamlit interface, the system supports three input methods: text entry, WAV file uploads, and live audio recordings. All machine learning models (Logistic Regression, DistilBERT, Wav2Vec2) were created and trained from scratch, delivering real-time emotion predictions. The system is intended for academic demonstration, research, and potential applications in sentiment analysis, mental health, and human-computer interaction.

**1.3 Definitions, Acronyms, and Abbreviations**

* **AI**: Artificial Intelligence
* **ML**: Machine Learning
* **TF-IDF**: Term Frequency-Inverse Document Frequency
* **DistilBERT**: A lightweight transformer model for natural language processing
* **Wav2Vec2**: A model for speech emotion recognition
* **Streamlit**: A Python framework for building web-based interfaces
* **WAV**: Waveform Audio File Format
* **GUI**: Graphical User Interface
* **SRS**: Software Requirements Specification

**1.4 References**

* Python Documentation: <https://www.python.org/doc/>
* Streamlit Documentation: <https://docs.streamlit.io/>
* Scikit-learn Documentation: <https://scikit-learn.org/stable/>

**2. Overall Description**

**2.1 Product Perspective**

The Emotion Recognition System is a standalone web application built from scratch, integrating custom-trained ML models for emotion detection. It operates as a client-side system, processing inputs locally or on a user-hosted server via Streamlit. The system does not rely on external APIs or pre-trained models, emphasizing original model development.

**2.2 Product Functions**

* Accept text input and predict emotions using Logistic Regression and DistilBERT models.
* Process WAV audio file uploads to detect emotions using the Wav2Vec2 model.
* Record live audio via microphone, generate temporary WAV files, and predict emotions.
* Display real-time emotion predictions through a Streamlit-based GUI.
* Ensure temporary audio files are deleted after processing for efficiency.

**2.3 User Characteristics**

* **End Users**: Students, researchers, or professionals interested in emotion analysis, with basic computer literacy and no prior ML expertise required.
* **Developers**: Individuals with Python and ML knowledge who may extend the system.
* **Administrators**: Academic evaluators (e.g., teachers) reviewing the system for project assessment.

**2.4 Constraints**

* Audio inputs must be WAV format, mono, 16kHz.
* Live recording requires a functioning microphone and Sounddevice library.
* System performance depends on hardware (e.g., CPU/GPU with at least 4 GB RAM).
* Models are trained for English text and audio inputs only.

**2.5 Assumptions and Dependencies**

* **Assumptions**: Users have Python 3.8+ installed, a compatible microphone, and internet access for initial library installation.
* **Dependencies**: Python libraries (Streamlit, Scikit-learn, Transformers, Librosa, Sounddevice, Soundfile, NLTK, Pandas, NumPy).

**3. Specific Requirements**

**3.1 Functional Requirements**

* **FR1: Text Input Processing**
  + The system shall accept a text sentence via a Streamlit text input field.
  + It shall predict emotions (sadness, joy, love, anger, fear, surprise) using custom-trained Logistic Regression and DistilBERT models.
  + Results shall be displayed on the GUI within 2 seconds.
* **FR2: Audio File Upload**
  + The system shall allow users to upload WAV files (mono, 16kHz) via a Streamlit file uploader.
  + It shall predict emotions (neutral, calm, happy, sad, angry, fearful, disgust, surprised) using a custom-trained Wav2Vec2 model.
  + Results shall be displayed on the GUI within 5 seconds.
* **FR3: Live Audio Recording**
  + The system shall record audio for 5-10 seconds (user-selectable) via a microphone.
  + It shall generate a temporary WAV file, process it with the Wav2Vec2 model, and delete the file after analysis.
  + Results shall be displayed on the GUI within 5 seconds post-recording.
* **FR4: Emotion Display**
  + The system shall display predicted emotions and model details (e.g., “Logistic Regression: joy”) on the Streamlit interface.
* **FR5: Error Handling**
  + The system shall display error messages for invalid inputs (e.g., empty text, non-WAV files, missing microphone).

**3.2 Non-Functional Requirements**

* **NFR1: Usability**: The GUI shall be intuitive, requiring no training for end users.
* **NFR2: Performance**: The system shall process inputs and display results within 5 seconds on standard hardware (4 GB RAM, 2.0 GHz CPU).
* **NFR3: Portability**: The system shall run on Windows, macOS, or Linux with Python 3.8+.
* **NFR4: Maintainability**: Code shall be modular, with comments for easy updates.
* **NFR5: Reliability**: The system shall handle 95% of valid inputs without crashing.

**3.3 External Interface Requirements**

* **User Interfaces**: Streamlit-based web GUI with text input fields, file upload buttons, sliders for recording duration, and result display areas.
* **Hardware Interfaces**: Microphone for live recording, compatible with Sounddevice library.
* **Software Interfaces**: Python 3.8+, Streamlit, Scikit-learn, Transformers, Librosa, Sounddevice, Soundfile, NLTK, Pandas, NumPy.
* **Communication Interfaces**: None; the system runs locally without external network requirements.

**4. System Features**

* **Text Emotion Analysis**: Processes user-entered text to predict emotions using custom-trained Logistic Regression and DistilBERT models, supporting emotions like joy, fear, and surprise.
* **Audio File Emotion Analysis**: Analyzes uploaded WAV files to detect emotions using a custom-trained Wav2Vec2 model, supporting emotions like happy and calm.
* **Live Audio Emotion Analysis**: Records audio via microphone, processes it with Wav2Vec2, and deletes temporary files, supporting real-time emotion detection.
* **Real-Time Feedback**: Displays emotion predictions instantly on a Streamlit GUI.
* **Error Management**: Provides clear error messages for invalid inputs or system issues.

**5. Other Non-Functional Requirements**

**5.1 Performance Requirements**

* Text predictions shall complete in under 2 seconds.
* Audio predictions (file or live) shall complete in under 5 seconds.
* System shall support up to 10 concurrent input processes on standard hardware.

**5.2 Security Requirements**

* No user authentication required, as the system is for local or demo use.
* Temporary audio files shall be deleted immediately after processing to ensure data privacy.

**5.3 Quality Attributes**

* **Scalability**: System can be extended to support additional input formats (e.g., MP3) with model retraining.
* **Robustness**: Handles invalid inputs gracefully with user-friendly error messages.
* **Accessibility**: GUI uses high-contrast colors and clear fonts for readability.

**6. Appendix**

* **Glossary**:
  + **Emotion Prediction**: The process of identifying human emotions from text or audio.
  + **Custom-Trained Model**: A machine learning model developed and trained from scratch by the developer.
* **Additional Notes**:
  + The system assumes English-language inputs for text and audio.
  + Future enhancements may include support for additional audio formats or languages.

Use Case Diagram

