**Software Requirements Specification (SRS)**

**Lie Detection Using Speech Processing Techniques**

**1. Introduction**

**1.1 Purpose of this Document**

This Software Requirements Specification (SRS) document is designed to serve as a comprehensive guide for the development of a lie detection system using speech processing techniques. The system employs advanced machine learning and deep learning methodologies to analyze speech features such as acoustic, prosodic, and emotional patterns for detecting deception. The purpose of this document is to outline the functional and technical requirements of the project, providing a detailed framework to guide the development team in creating an efficient and accurate lie detection system. This document will ensure all stakeholders have a clear understanding of the project objectives, constraints, and deliverables while navigating the complexities of speech analysis and machine learning integration.

**1.2 Scope of The Project**

The primary focus of this project is to develop a system capable of detecting deception by analyzing speech patterns. The system utilizes machine learning algorithms (e.g., Support Vector Machines, Decision Trees) and deep learning techniques (e.g., CNN, LSTM) to extract and process features from speech, such as pitch, intonation, and emotional cues.

The project involves creating a robust pipeline for speech preprocessing, feature extraction, and model training, allowing the system to analyze speech data accurately and efficiently. The application will target various domains, including security, law enforcement, and psychological research, providing a non-invasive alternative to traditional lie detection methods. The software will support real-time analysis, leveraging pre-trained models for high accuracy and scalability.

Developed in Python, the project will integrate key libraries and tools such as Librosa for audio analysis and TensorFlow/PyTorch for machine learning and deep learning tasks. The deliverables include a web-based user interface for input and output interaction, datasets for training and testing, and model evaluation reports to ensure transparency and reliability.

# **2. General Description**

The project entails developing a speech-based lie detection system that leverages cutting-edge speech processing techniques and AI models. The system will analyze acoustic (e.g., pitch, speech rate), prosodic (e.g., rhythm, intonation), and emotional (e.g., stress, anxiety) features in speech to detect deceptive behavior.

Machine learning models, including both traditional (e.g., SVM, Decision Trees) and advanced deep learning architectures (e.g., CNN, LSTM), will be employed to achieve high detection accuracy. Speech data will be preprocessed to extract key features, which will then be fed into models for training and evaluation.

A web-based user interface will allow users to upload audio recordings and view the results of lie detection analysis in real-time. The system will also include visualization tools to display key patterns and insights derived from speech data. By incorporating datasets such as the Columbia SRI-Colorado Corpus and the Deceptive Speech Corpus, the project aims to ensure robust model training and testing.

Developed on Python, the system will be deployable across multiple platforms and environments, ensuring scalability and versatility for real-world applications.

## **2.1 Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **AI (Artificial Intelligence)** | A branch of computer science focused on creating systems capable of performing tasks requiring human intelligence. |
| **ML (Machine Learning)** | A subset of AI where algorithms learn patterns from data to make predictions or decisions. |
| **DL (Deep Learning)** | A subset of ML that uses neural networks with many layers to analyze complex patterns in data. |
| **Speech Processing** | Techniques used to analyze and process speech signals for various applications, such as recognition or analysis. |
| **Acoustic Features** | Characteristics of sound, such as pitch, intensity, and formant frequency, used in speech analysis. |
| **Prosodic Features** | Speech attributes related to rhythm, intonation, and stress, providing insights into cognitive and emotional states. |
| **Emotion Recognition** | The process of identifying emotions in speech or other forms of data through analysis and modeling. |
| **LSTM (Long Short-Term Memory)** | A type of recurrent neural network (RNN) that is effective in processing sequential data, such as speech signals. |
| **SVM (Support Vector Machine)** | A machine learning algorithm used for classification and regression tasks by finding an optimal decision boundary. |
| **Librosa** | A Python library for analyzing and extracting features from audio signals |
| **TensorFlow/PyTorch** | Popular frameworks for implementing and training machine learning and deep learning models. |
| **Dataset** | A collection of data samples used for training and testing machine learning models. |
| **User** | An individual interacting with the lie detection system, typically providing audio input for analysis. |

## **2.2 User Characteristics**

* Technical Proficiency: Users are not required to have in-depth knowledge of machine learning or speech processing. However, familiarity with basic computing concepts and the ability to upload audio files to the system is necessary.
* Professional Background: The primary users of this system may include professionals in law enforcement, psychological research, and security. These users may not have technical expertise but are expected to interpret the system's results based on provided insights.
* Training Requirements: Minimal training is required for users to interact with the web-based interface. The system will include a help guide and tooltips for user assistance.
* Problem-Solving Skills: Users should be capable of applying the system's outputs to their specific contexts, such as investigations, behavioral studies, or risk assessments.
* Adaptability: Users must be open to adopting this new technology as a supplementary tool for detecting deception, which may involve integrating it with existing workflows.

## **2.3 Overview of Functional Requirements**

* Speech Analysis: The system will analyze audio input files to extract relevant acoustic, prosodic, and emotional features.
* Lie Detection: Employ machine learning and deep learning models to classify speech as deceptive or truthful based on the extracted features.
* Real-Time Processing: Enable real-time or near-real-time analysis for scenarios requiring immediate results, using pre-trained models.
* User Interface: Provide a user-friendly web interface for audio input, displaying results, and accessing visualization tools that explain the analysis.
* Visualization Tools: Include graphical representations of speech patterns, highlighting features contributing to deception detection (e.g., pitch variations, stress markers).
* Model Customization: Allow for updates or retraining of models with new datasets to improve system accuracy and adapt to evolving requirements.
* Data Security: Ensure that user data, including uploaded audio files, is handled securely and complies with relevant privacy standards.

## **2.4 General Constraints and Assumptions**

**Constraints**:

1. **Hardware Requirements**: The system's performance may depend on the server's computational capabilities for running deep learning models and processing speech data.
2. **Dataset Availability:** The quality and diversity of training datasets directly influence the system's accuracy and reliability.
3. **Real-Time Limitations:** Real-time processing may be constrained by the size and complexity of the audio files and the computational resources available.
4. **Internet Dependency:** The system requires a stable internet connection for users to access the web interface and perform analyses.

**Assumptions**:

1. **User Proficiency:** It is assumed that users will have basic computer skills and access to devices capable of recording or uploading audio files.
2. **Speech Quality:** The audio input provided will be of sufficient quality (e.g., clear speech with minimal background noise) to enable effective analysis.
3. **Ethical Use:** It is assumed that the system will be used ethically and responsibly within legal and organizational guidelines.
4. **Model Accuracy:** The system's accuracy is dependent on the training data and model optimization, assuming that the provided datasets are comprehensive and unbiased.

# **3. Specific Requirements**

## **3.1 Interface Requirements**

### **3.1.1 User Interface**

* **Real-Time Data Visualization:**  
  Implement a user-friendly interface to visualize real-time analysis results, including acoustic, prosodic, and emotional features detected in the speech input.
* **Interactive Dashboard:**  
  Provide a dashboard displaying analysis metrics such as deception probability, emotional states, and acoustic feature changes, allowing users to monitor the system's performance and insights.
* **Audio Upload Interface:**  
  Design a simple audio file upload interface for users to input their recordings for analysis.

### **3.1.2 Hardware Interface**

* **Computing Power:**  
  Ensure adequate computing resources, including high-performance CPUs and GPUs, to process complex speech features and run machine learning models efficiently.
* **Audio Input Devices:**  
  Support for standard audio recording devices such as microphones for real-time audio capture.
* **Peripheral Support:**  
  Compatibility with basic peripherals like monitors and input devices (keyboard, mouse) for interacting with the user interface.

### **3.1.3 Software Interface**

* **Python Environment:**  
  Compatibility with Python and necessary libraries (e.g., Librosa, TensorFlow, PyTorch) for audio processing and model execution.
* **Model Integration:**  
  Seamless integration of machine learning models (SVM, CNN, LSTM) for feature extraction and lie detection analysis.
* **Database Support:**  
  Integration with a database system (e.g., PostgreSQL) for storing analysis results and logs.
* **Operating System Support:**  
  Ensure system compatibility with major operating systems like Ubuntu and Windows.

### **3.1.4 Communication Interfaces**

* **Data Exchange Protocols:**  
  Implement secure and efficient data exchange protocols for audio files and model outputs.
* **API Integration:**  
  Develop APIs for connecting the system with external services or other software components for data sharing and model updates.
* **Real-Time Communication Support:**  
  Ensure communication channels can handle real-time audio processing and results generation with minimal latency.

## **3.2 Detailed Description of Functional Requirements**

### **3.2.1 Template for Describing Functional Requirements**

For each functional requirement of the lie detection system, the following template will be used:

* **Requirement Name:** A clear and concise name for the requirement.
* **Description:** A detailed description of the requirement, explaining its purpose within the system.
* **Priority:** The level of importance of the requirement (e.g., High, Medium, Low).
* **Inputs:** The inputs required for the requirement to function (e.g., audio files, user commands).
* **Processing Steps:** A step-by-step explanation of how the requirement will be processed within the system.
* **Outputs:** The expected outputs or results from the requirement (e.g., deception probability, emotional state analysis).
* **Dependencies:** Any other system components or requirements that this requirement relies on to function properly.

## **3.3 Non-Functional Requirements**

**1. Performance**

* The system must process speech data and generate analysis results within 1000 milliseconds of receiving input, ensuring real-time feedback with latency not exceeding 2000 milliseconds under normal operating conditions.

**2. Reliability**

* The system must maintain a 99% uptime and demonstrate consistent performance when handling varied speech inputs and concurrent user requests, supporting up to 100 simultaneous users without degradation.

**3. Usability**

* The user interface must have a System Usability Scale (SUS) score of 80 or higher, enabling non-technical users to perform key tasks (e.g., uploading speech files, viewing results) within 3 clicks or less.

**4. Scalability**

* The system must scale to handle a 50% increase in data load (e.g., from 100 GB to 150 GB) and support up to 100 concurrent users with response times remaining under 1500 milliseconds.

**5. Maintainability**

* The system architecture must be modular, allowing for independent updates to components without affecting the rest of the system. All code must include unit tests with at least 90% coverage and detailed documentation of key modules.

**6. Security**

* Implement data encryption using AES-256 for storage and TLS 1.3 for transmission. The system must enforce multi-factor authentication (MFA) and log all access attempts, retaining logs for a minimum of 90 days.

**7. Compatibility**

* The system must be compatible with at least 90% of commonly used hardware and software configurations, including Windows, macOS, Linux, and a variety of microphones (e.g., USB and analog).

**8. Portability**

* Trained models must be exportable in standard formats (e.g., ONNX or TensorFlow SavedModel) and deployable on different platforms (e.g., cloud environments, edge devices) with no more than 10% performance degradation.

**9. Compliance**

* The system must comply with GDPR and CCPA regulations for data privacy, and adhere to ethical AI guidelines as defined by ISO/IEC 22989 (AI ethical frameworks).

**10. Accessibility**

* Ensure conformance to WCAG 2.1 Level AA guidelines, including features like real-time captioning for speech analysis results and keyboard navigation for all system functionalities.

# **4. Analysis-UML**

## **4.1 Use Cases**

### **4.1.1 Use Case Diagram**

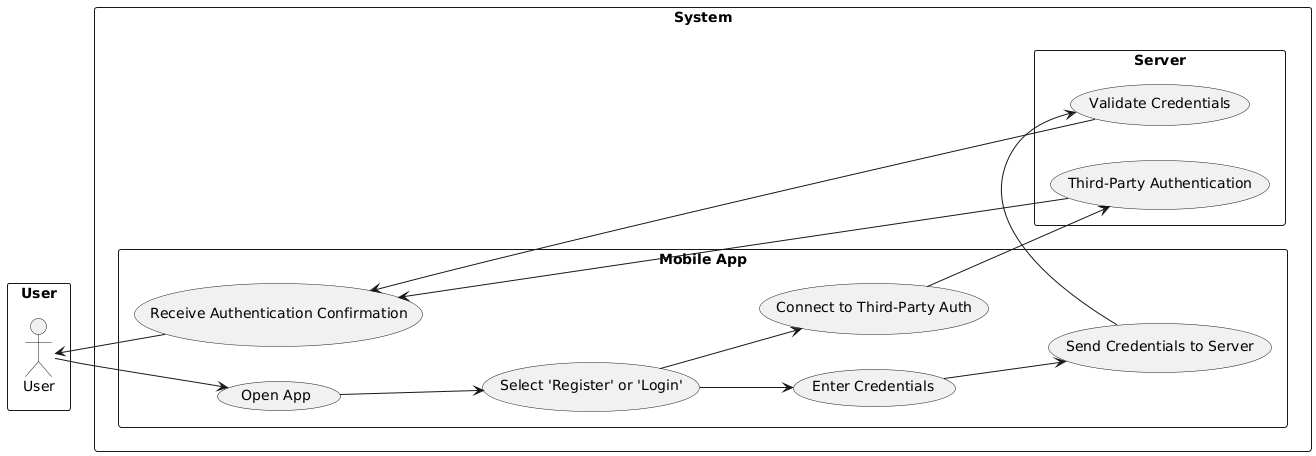
****

Diagram 1 - Use Case 1 Diagram

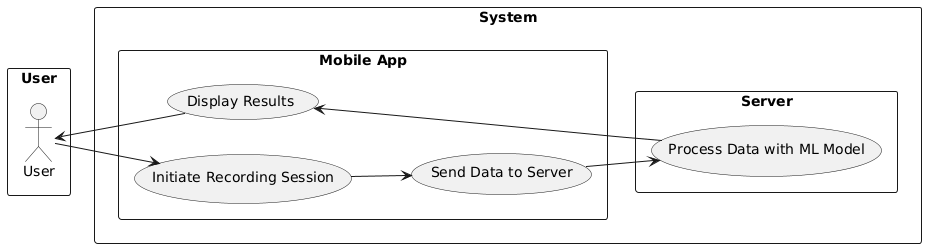


Diagram 2 - Use Case 2 Diagram

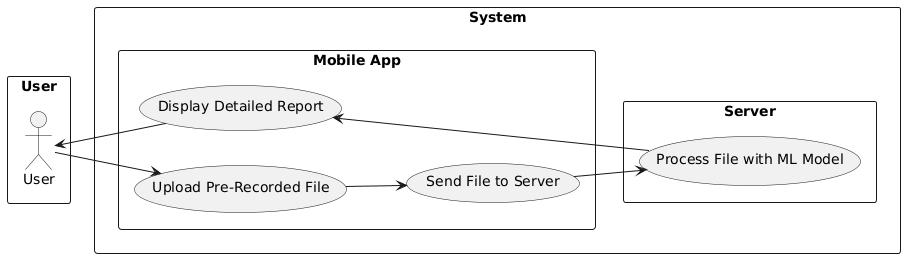


Diagram 3 - Use Case 3 Diagram

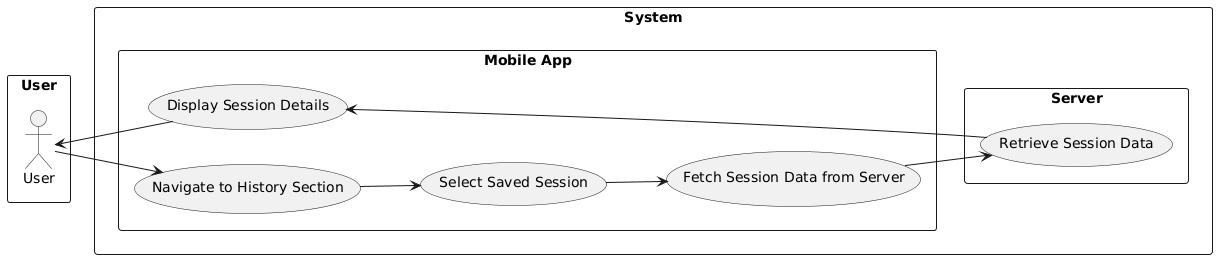


Diagram 4 - Use Case 4 Diagram

### **4.1.2 Describe Use Cases**

|  |  |
| --- | --- |
| Field | Description |
| Use Case Number | Use Case 1 |
| Use Case Name | User Registration and Authentication |
| Actor | User |
| Description | Users create an account or log in to access the app. |
| Precondition | The user must have the app installed and an active internet connection. |
| Scenario | 1. User opens the app. |
|  | 2. Selects "Register" or "Login." |
|  | 3. Enters credentials (email/phone, password) or uses third-party authentication (e.g., Google, Apple ID). |
|  | 4. The app sends credentials to the server for validation or connects to a third-party authentication service. |
|  | 5. The server or third-party provider verifies the credentials. |
|  | 6. The app receives a confirmation of successful authentication. |
| Postcondition | The user is authenticated and granted access to app features. |
| Exceptions | Invalid credentials entered, third-party authentication unavailable, or server issues prevent successful authentication. |

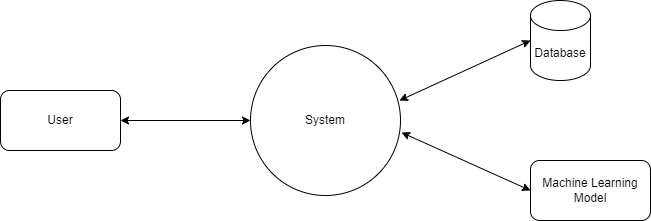
|  |  |
| --- | --- |
| Field | Description |
| Use Case Number | Use Case 2 |
| Use Case Name | Real-Time Lie Detection |
| Actor | User |
| Description | The user records or streams audio/video, and the app sends the data to the server for lie detection. |
| Precondition | The user must grant microphone/camera permissions.  The device must be connected to the server. |
| Scenario | 1. The user initiates a recording session. |
|  | 2. The app sends the recorded data to the server. |
|  | 3. The server processes the data using the machine learning model. |
|  | 4. The app displays the result: "Truth," "Lie," or "Uncertain." |
| Postcondition | Results are displayed to the user. |
| Exceptions | Microphone or camera permissions not granted, Server connectivity issues, Errors during the server-side ML model processing. |

|  |  |
| --- | --- |
| Field | Description |
| Use Case Number | Use Case 3 |
| Use Case Name | Analysis of Recorded Conversations |
| Actor | User |
| Description | The user uploads a pre-recorded audio or video file for analysis. |
| Precondition | The file format is supported. The device is connected to the server. |
| Scenario | 1. The user uploads a file. |
|  | 2. The app sends the uploaded data to the server. |
|  | 3. The server processes the data using the machine learning model. |
|  | 4. The app provides a detailed report on potential lies and truthfulness. |
| Postcondition | Results are displayed to the user. |
| Exceptions | Unsupported file format, Server connectivity issues, Errors during the server-side ML model processing. |
|  |  |
| Field | Description |
| Use Case Number | Use Case 4 |
| Use Case Name | View History and Reports |
| Actor | User |
| Description | Users can view past lie detection reports and session summaries. |
| Precondition | The user must be authenticated, The user must have prior saved sessions available in the app. |
| Scenario | 1. The user navigates to the "History" section. |
|  | 2. The user selects a session from the list of saved sessions. |
|  | 3. The app displays detailed session information. |
| Postcondition | The user gains insights and can review the analysis of previous sessions. |
| Exceptions | No saved sessions available for the user, Errors retrieving session data from the server. |

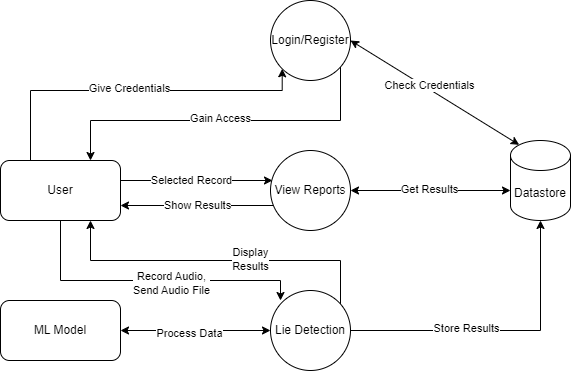
## **4.2 Functional Modeling (DFD)**

### **4.2.1 DFD Diagrams**

**Level-0 Diagram**

****

**Level-1 Diagram**

****

# **Conclusion**

The "Lie Detection Using Speech Processing Techniques" project aims to provide a fast, accurate, and non-invasive method for detecting deception through speech analysis. By utilizing advanced AI and speech processing technologies, the system offers practical applications in fields like law enforcement, security, and psychological research.

This project focuses on creating a reliable and user-friendly tool that can analyze speech patterns in real time, providing clear and actionable results. With its ability to adapt to new data and evolving needs, the system has the potential to become a valuable resource for professionals seeking a deeper understanding of truth and trust in various contexts.

# **References and Resources**

 **Librosa Documentation**  
<https://librosa.org/>  
A Python library for audio and speech feature extraction.

 **TensorFlow Documentation**  
<https://www.tensorflow.org/>  
A popular deep learning framework used for implementing and training machine learning models.

 **PyTorch Documentation**  
<https://pytorch.org/>  
A flexible and efficient deep learning framework for building neural networks.

 **Columbia SRI-Colorado Corpus**  
A dataset of speech samples widely used for research in speech processing and emotion recognition.

 **Deceptive Speech Corpus**  
A specialized dataset for studying speech-based deception detection.