SOFTWARE REQUIREMENT SPECIFICATIONS

For Vocal Frequency Recognition Of Birds

23-Oct-2004

Prepared by

Specialization	**SAP ID**	**Name**
Data Science	500108000	Somya Yadav
Data Science	500108594	Mansha Chaudhary
Data Science	500105016	Roshita Ailawadi

MENTOR SIGNATURE

School Of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN- 248007. Uttarakhand

TABLE OF CONTENTS

Revision History

Date	Change	Reason for Changes	Mentor Signature

1. INTRODUCTION

1.1 Purpose of the Project

This project aims to analyze the vocal frequency patterns of the Jacobin cuckoo, a bioindicator species, and correlate them with climate and environmental factors. This addresses the problem of how climate change affects the bird's behavior, specifically its vocalizations, migration, and breeding habits. The project is motivated by the need to monitor biodiversity changes in response to climate impacts, aiding in conservation efforts.

1.2 Target Beneficiary

The prime beneficiaries are **conservationists**, **ecologists**, and **climate researchers**, who can use the findings to understand the ecological impact of climate change and inform strategies for biodiversity protection.

1.3 Project Scope

The project aims to generate and analyze spectrograms of the Jacobin cuckoo's vocalizations, correlating these patterns with environmental data such as temperature, precipitation, and seasonal changes. By examining how climate factors influence the bird's vocal behaviour, the project provides valuable insights into the effects of climate variability on the species. This information is crucial for ecological monitoring, enabling researchers to detect shifts in the cuckoo's behaviour that may signal broader environmental changes. The project deliverables include:

Spectrograms displaying vocal frequency patterns.

Data correlation analysis linking vocal changes with environmental variables.

Conservation insights to support strategies for protecting the species and its habitat.

1.4 References

- A review of automatic recognition technology for bird vocalizations in the deep learning era - ScienceDirect

- **Spectrogram in JavaScript (arc.id.au)**

- **Audio Data Processing— Feature Extraction — Essential Science & Concepts behind them — Part I | by Vasanthkumar Velayudham | | Analytics Vidhya | Medium**

- **java - Using FFT to compare two audio files and then realtime comparison - Stack Overflow**

2.PROJECT DESCRIPTION

2.1 Reference Algorithm

The reference algorithm for this project is based on **Short-Time Fourier Transform (STFT)**, a well-known method for frequency domain analysis. Here's an outline of the algorithm:

1. **Data Collection**:

   Audio recordings of the Jacobin cuckoo's vocalizations are collected over several years, from different geographical locations, covering a wide range of environmental conditions.

2. **Preprocessing (STFT)**

   Preprocessing for the Short-Time Fourier Transform (STFT) in your vocal frequency recognition project involves several key steps. First, the continuous audio signal of the Jacobin cuckoo is divided into short, overlapping frames (typically 20-40 ms), assuming the signal remains stationary within each frame. Next, each frame is multiplied by a window function, such as Hamming, to minimize edge effects and ensure smooth transitions between frames. Then, the Fast Fourier Transform (FFT) is applied to each windowed frame, converting the time-domain audio into the frequency-domain

   Finally, the frequency data from all frames are combined to generate a spectrogram, which visually represents how vocal frequencies vary over time. This preprocessing enables the analysis of dominant frequencies and temporal patterns in the bird's vocalizations, allowing for insights into environmental impacts on their behaviour.

3. **Feature Extraction**:

   Once the spectrogram is generated, key features are extracted using algorithms in Java:

   Dominant Frequencies: Identify the peak frequencies in each audio segment.

   Energy Distribution: Analyze how energy is distributed across different frequency bands to detect changes in vocal intensity.

   Temporal Changes: Track variations in these features over time to understand shifts in vocalization behaviour.

4. **Correlation with Environmental Data**:

   Historical weather data, such as temperature, precipitation, and humidity, is collected and correlated with the vocal frequency patterns. Use statistical analysis techniques to identify any relationships between environmental changes and the bird's vocal behaviour.

5. **Pattern Recognition**:

   Machine learning models, such as decision trees or neural networks, can be trained to recognize patterns in the vocal frequencies that correspond to different environmental conditions

Required Data Structure

For handling and analysing this data, the following data structures can be used:

Time-Series Data Structure: Since you're analysing vocal frequencies over time, a time-series data structure is ideal for storing both the audio frequency data and the weather data (e.g., timestamps, frequency values, weather conditions).

Matrix Data Structure for FFT: Spectrograms can be represented using a matrix where each element corresponds to the intensity of a specific frequency at a particular time.

Justification for the methodology

The STFT algorithm is particularly suitable for this project as it allows for efficient and real-time analysis of vocal frequency patterns, which is crucial for understanding how the Jacobin cuckoo's behaviour responds to environmental changes. The integration of machine learning models enhances the ability to identify subtle patterns in vocalizations that may correlate with climate factors, providing insights valuable for conservation efforts.

2.2 Characteristics of Datasets

The dataset used in our vocal frequency recognition project comprises two primary components:

1. **Audio Recordings**:

   Type: Time-series audio data capturing the vocalizations of the Jacobin cuckoo.

   Duration: Recordings span from the year 2000 to 2024, capturing the bird's vocal behavior across different geographical regions and time periods.

   Features: Frequency data extracted from the recordings, including dominant frequencies, energy distributions, and temporal changes.

   Sampling Rate: High-quality audio files sampled at appropriate rates (typically 44.1 kHz or higher) to ensure clear representation of the bird's vocal frequencies.

2. **Environmental Data**:

   Type: Time-series weather data corresponding to the periods when audio recordings were made.

   Parameters: Temperature, precipitation, humidity, and other relevant weather factors.

   Temporal Resolution: Daily or hourly data points depending on availability, to match the temporal resolution of the vocalization data.

Primary And Seconadary Sources of Data

Primary Source:

Field Recordings: Audio data collected from field recordings of the Jacobin cuckoo. These recordings are either sourced from ongoing research projects or specific biodiversity observation programs.

Historical Weather Data: Sourced from official meteorological agencies such as the **Indian Meteorological Department (IMD)**, or other trusted global databases like **NOAA** (National Oceanic and Atmospheric Administration) for weather statistics relevant to the periods when the recordings were made.

Secondary Source:

Public Audio Datasets: Supplementary audio data may be sourced from publicly available ornithology databases, such as **Xeno-canto** (an extensive database of bird sounds).

Climate and Environmental Data Archives: Secondary climate data can be obtained from online repositories like **WorldClim** or **ERA5**, which provide access to global climate datasets.

Sampling Techniques

1. **Audio Data Sampling**:

   Stratified Sampling: The audio recordings are stratified based on geographical regions, seasons, and time periods to ensure a representative sample that captures the full range of the Jacobin cuckoo's vocal behavior across different environmental conditions.

   Systematic Sampling: Regular intervals of vocalizations within the recording are sampled to ensure consistent analysis across different time points.

2. **Weather Data Sampling**:

   Temporal Sampling: Weather data is matched to the audio recording timestamps, ensuring temporal alignment for accurate correlation between the vocalization and environmental factors.

   Spatial Sampling: Weather data is sampled from meteorological stations or grids that are closest to the cuckoo's habitat to accurately reflect local environmental conditions.

Statistical Method for Data Processing

Correlation Analysis:

Pearson Correlation: This statistical method is used to measure the strength and direction of the linear relationship between the vocal frequencies of the Jacobin cuckoo and environmental factors (e.g., temperature, precipitation). It helps identify which weather variables have the strongest influence on vocalization patterns.

Time-Series Analysis:

Moving Average: A moving average is applied to smooth out fluctuations in vocalization patterns over time, helping to detect longer-term trends in the data.

FFT (Fast Fourier Transform):

As mentioned earlier, FFT is used to convert the audio data from the time domain into the frequency domain, allowing for the extraction of dominant frequencies and their variations over time.

Regression Analysis:

Linear Regression: Used to predict the changes in vocal frequency based on the input environmental variables. This statistical method helps model how specific climate factors, such as temperature or precipitation, might predict vocal behavior.

2.3 Swot Analysis

Strengths:

Ecological Relevance: The Jacobin cuckoo is a key bioindicator of seasonal and climate changes, offering real-time insights into environmental impacts.

Innovative Approach: The project combines vocal frequency analysis with weather data, integrating data science and ecology for conservation research.

Scalability: The methodology is adaptable to other species and regions, enhancing its future potential.

Conservation Impact: Findings contribute directly to conservation strategies by highlighting the effects of climate change on the cuckoo's behavior.

Weaknesses:

Data Availability: Limited access to long-term, geographically varied audio and weather data can affect accuracy.

Technical Complexity: Requires advanced computational resources and expertise, making accurate data processing challenging.

Environmental Variability: Non-climate factors like habitat disturbance can complicate data analysis.

Opportunities:

Advances in Machine Learning: Future models can improve accuracy in analyzing vocal patterns.

Global Climate Research: The project can provide valuable data on biodiversity and climate impact.

Expandability: The approach can be applied to other species for broader ecological monitoring.

Public Awareness: Findings can promote conservation through education and engagement.

Threats:

Climate Unpredictability: Rapid environmental changes could disrupt predictions and require constant model updates.

Technological Limitations: Emerging technologies could make the current methods obsolete.

Dependence on External Data: Gaps in weather or audio data could compromise the project's scope.

Conservation Policy: Implementation relies on external political and financial support.

2.4 Project Features

The project focuses on understanding the vocal behavior of the Jacobin cuckoo and how it correlates with environmental changes, particularly weather patterns. Key features include:

1. **Audio Data Collection**: The system will gather audio recordings of the Jacobin cuckoo from 2000 to 2024. This historical dataset will provide a rich basis for analysis, capturing variations in vocalizations over two decades.

2. **Weather Data Correlation**: Although users cannot input or link weather data directly, the project will utilize external datasets to correlate bird vocalizations with environmental factors. The analysis aims to identify how variations in weather, such as temperature and rainfall, impact the Jacobin cuckoo's calls and migration patterns.

3. **Spectrogram Generation**: The project will include a feature to generate spectrograms from the collected audio files. These visual representations will help in analyzing the frequency and duration of the cuckoo's calls, providing insights into how these vocalizations change in response to different environmental conditions.

4. **Vocalization Analysis**: The project will employ machine learning algorithms to analyze the extracted features of the vocalizations. By examining changes in pitch, frequency, and call length, the project aims to reveal significant trends related to the cuckoo's behavioral adaptations to climate variability.

5. **Migration Pattern Insights**: By synthesizing the analysis of vocalizations with weather data, the project will explore the relationship between environmental factors and the Jacobin cuckoo's migration. This feature will help determine how climate change affects the bird's migration routes and timings, enhancing understanding of its ecological role.

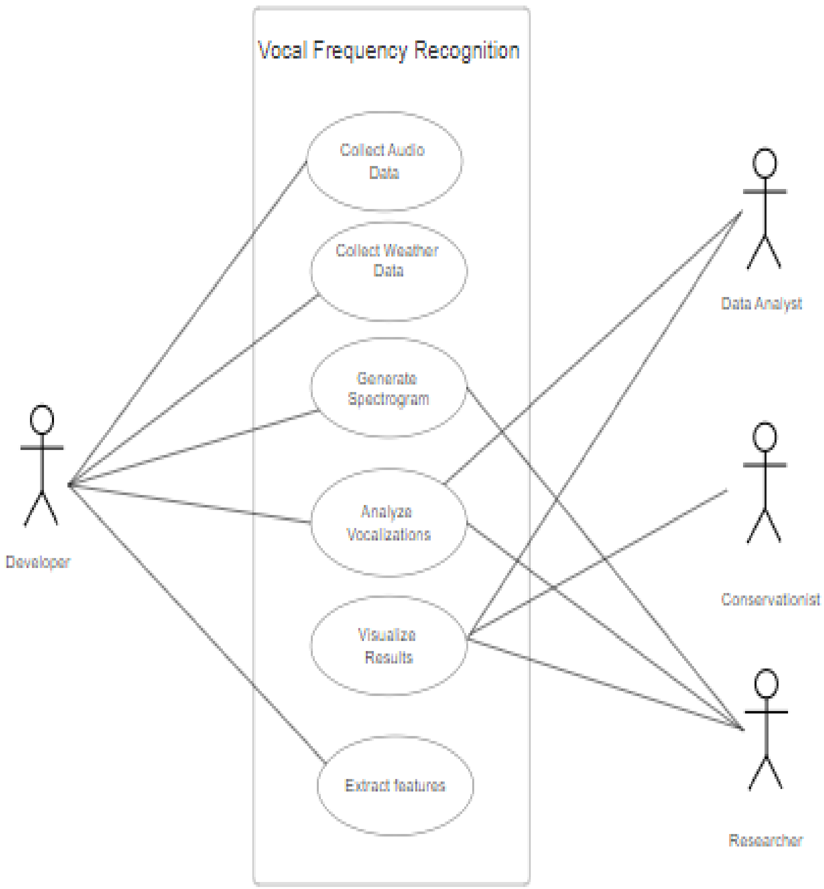
6. **Visualization Tools**: The system will include tools to visualize the relationships between vocalizations and environmental changes. Graphs and charts will illustrate

patterns and trends, aiding in comprehending how external factors influence the cuckoo's behavior.

7. **User-friendly Interface**: While direct input of weather data is not allowed, the project will provide a user-friendly interface for viewing the analysis results. Users can easily navigate through audio files, spectrograms, and visualized data correlations.

These features collectively aim to create a comprehensive understanding of the Jacobin cuckoo's vocalizations, linking them with environmental changes and ultimately contributing to conservation efforts by highlighting the impacts of climate change on wildlife.

USE CASE



2.5 User Classes and Characteristics

The vocal frequency recognition system will cater to multiple user classes, each with unique characteristics and requirements:

1. Researchers and Ecologists:

These users focus on studying the Jacobin cuckoo's vocalizations in relation to environmental factors. They require detailed analytical tools, including spectrogram visualizations and data correlation capabilities. They value accuracy and depth of analysis, as their research may contribute to conservation strategies.

2. Wildlife Enthusiasts:

This user class includes bird watchers and amateur ornithologists interested in understanding bird behavior through vocalizations. They need a user-friendly interface that allows them to upload audio recordings and receive basic analyses without requiring extensive technical knowledge.

3. Environmental Scientists:

These professionals assess the impact of environmental changes on wildlife. They require comprehensive reports on vocal frequency changes correlated with weather patterns and habitat disturbances. They value functionalities that facilitate data export and presentation of findings.

4. Conservation Organizations:

These users utilize the system for advocacy and policymaking. They need summarized insights into how environmental changes affect bird populations, including trends that could inform conservation efforts. They prioritize features that provide clear visualizations and actionable data.

5. Software Developers and Data Analysts:

These users may contribute to the system's development or perform advanced analyses on the data collected. They require access to raw data, documentation for further development, and tools for custom analysis. They are tech-savvy and prioritize flexibility and extensibility in the software.

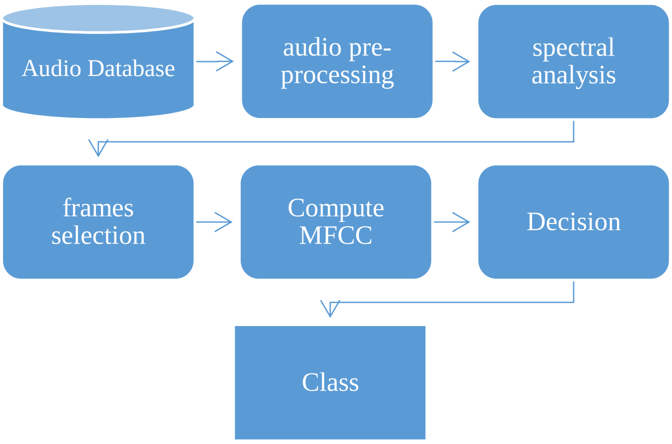
2.6 Design and Implementation Constraints

The following constraints can be considered while developing the system:

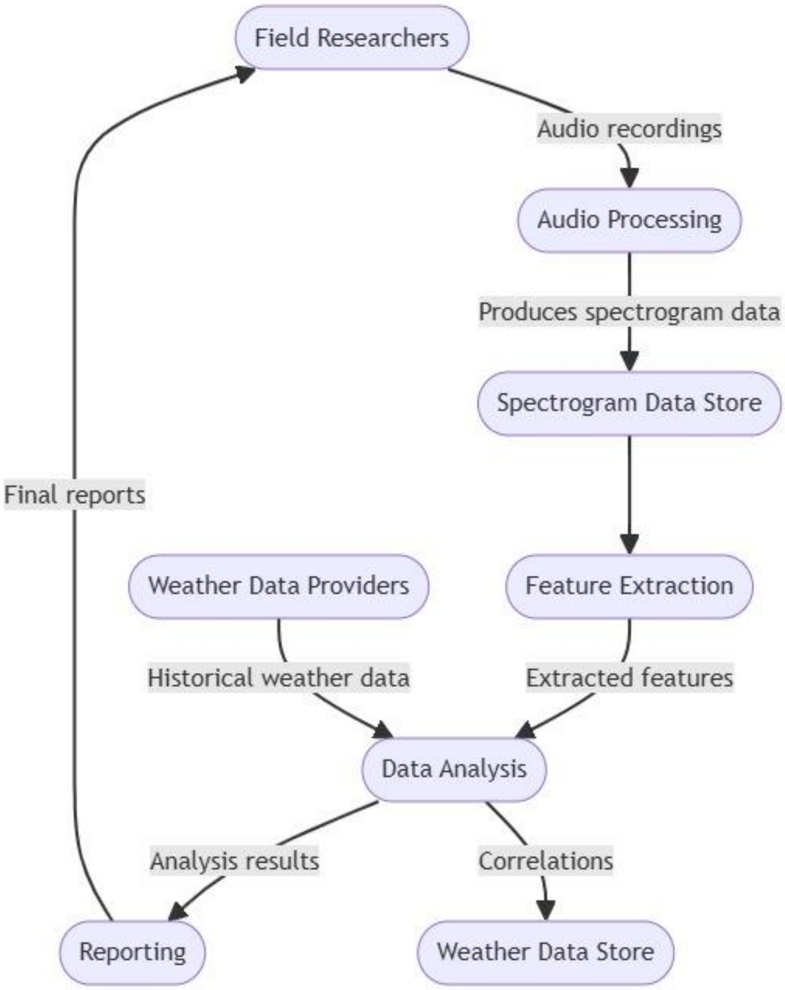
1. **Programming languages**:
   Java(used for backend/server-side) and Javascript (used for frontend/client-side) need to work seamlessly together for smooth interaction between the user interface and the backend processing.
2. **Audio processing libraries:**
   Java is used for audio processing using third-party libraries (eg. JAudioTagger for reading metadata or JTransform for Fourier transform). The libraries should be compatible with the specific audio formats(.wav or .mp3).
3. **Spectrogram Generation:**
   The system will rely on algorithms such as MFCC or STFT to generate spectrograms from audio files. So, the system must be optimized to provide results in real time.
4. **File format constraints**:
   The file formats supported by the system are .wav and .mp3. So the users should be given clear instructions and validation checks.
5. **User experience:**
   The UI must be designed for users without major technical knowledge. It should be easy for them to upload files, select constraints from the dropdown menu, and read the outputs generated.
6. **Cross-platform compatibility:**
   The platform needs to be compatible with various devices(desktops, smartphones, etc.).
7. **Security:**
   Data should be handled carefully to prevent unwanted content. The communications need to be secured with HTTPS.

2.7 Design Diagrams

1. **Class diagram:**



2, DFD



2.8 Assumptions and Dependencies

Assumptions:

1. audio files uploaded by the user are of adequate quality and in supported formats.
2. Audio files represent bird vocalizations and are suitable for analysis
3. Users have a stable internet connection for uploading audio files and receiving results in real-time.
4. Users have a basic knowledge of interacting with the interface.

Dependencies:

1. The backend is developed in Java and depends on its ability to process large audio files efficiently.
2. Modern browsers such as Chrome, Firefox, Safari, etc. are used for accessing the web interface.
3. For secure file upload handling, proper integration with Java's file I/O libraries and frontend validation (Javascript).

3. SYSTEM REQUIREMENTS

3.1 User interface- The platform through which users can interact with the system will be designed with simplicity, usability, and responsiveness. The following features will be added to the UI:

1. **File upload:** A 'choose file' option for the user to upload the audio file(.wav or .mp3 format).
2. **Dropdowns:** various dropdown options for the user to choose various features like year, location, weather, etc. based on which spectrograms will be generated for comparison.
3. **Spectrogram Display**: after the processing of audio files, spectrograms will be generated based on the features opted by the user for comparison of audio files.
4. **Real-time updates:** The system will dynamically display results without needing a page refresh. Users will be able to upload additional files and generate new spectrograms in real-time.
5. **Background Animations:** CSS animations integrated into the background will create a visually engaging experience and enhance the overall look of the interface.
6. **Responsiveness:** The interface will be compatible across different devices and screen sizes, offering a fluid experience.

3.2 System interface- The backend and the front end will combine to create the software interface, which includes communication protocols and file handling.

1. **Frontend:** Javascript is used to handle dynamic user interactions like animations for interface designing, handling file inputs, spectrogram generations, etc.
2. **Backend:** The backend will handle the processing of audio files, extracting relevant features and frequency diagrams using algorithms like Short-Time Fourier Transform (STFT) or MFCC.
3. **APIs/Libraries:** At the backend, Third-party libraries like STFT and MFCC for processing audio are used. At the front, Bootstrap libraries for a responsive UI, chart.js to display the spectrogram, window.AudioContext and window.webkitAudioContext for processing of audio files are used.

3.3 Database interface- The system doesn't require a traditional database, as it operates in real-time, and storage of user files or spectrograms is not required.

1. **Temporary storage:** Audio files uploaded by the user will be stored temporarily in a session-based or cache-based storage mechanism. After the generation and display of spectrograms, the files will be discarded automatically.
2. **Session management:** Each session will be managed separately, allowing users to upload multiple files without conflict.

3.4 Protocols-

1. **HTTP/HTTPS:** Standard HTTP for web communications and HTTPS will be used to ensure security and encrypt data transferred between the client and the server.
2. **File upload protocol**: the system will enforce secure file upload for handling file uploads. This includes:
    - Preventing overloading the server by limiting the file size
    - Validating file types (.wav and .mp3 formats will only be accepted).
    - Preventing uploading of unwanted or malicious files by scanning the files during the upload process.

4. NON - FUNCTIONAL REQUIREMENTS

4.1 Performance requirements

The performance of the vocal frequency recognition system for the Jacobin cuckoo is crucial to ensure accurate and timely analysis of audio data. The following performance requirements outline the benchmarks that the system must meet:

1. **Processing Time:** The system should be capable of processing audio recordings and generating frequency recognition results in a timely manner. Specifically, the system should process a one-minute audio recording and produce analysis results within 30 seconds. This quick turnaround is essential for researchers who may need real-time insights during field studies.
2. **Recognition Accuracy:** The system must achieve a recognition accuracy of at least 90% when identifying vocalizations of the Jacobin cuckoo from a variety of environmental backgrounds. This accuracy is critical for ensuring that the data collected is reliable and can be used in further research and conservation efforts. Continuous performance evaluation should be conducted using a validation dataset to confirm this benchmark.
3. **Scalability:** As the volume of audio data collected increases, the system should be capable of handling a growing dataset without significant degradation in performance. The architecture should support concurrent processing of multiple audio files, allowing simultaneous analyses without affecting the system's speed or accuracy.
4. **Resource Utilization:** The system should be optimized for efficient use of computational resources, including CPU and memory. Resource utilization should be monitored to ensure that the system can run on standard hardware configurations without requiring excessive processing power, making it accessible to a broader range of users.
5. **Data Storage and Retrieval:** The system should enable efficient storage and retrieval of audio recordings and analysis results. Query response times for retrieving specific recordings should not exceed 2 seconds, even with a database containing thousands of audio files. This ensures that users can access the data they need quickly.
6. **User Load Handling:** The system should be designed to support multiple users simultaneously, with the capability to handle at least 50 concurrent users without

performance degradation. This is particularly important for collaborative research projects involving multiple stakeholders.

7. **Reporting and Visualization Speed:** Generated visualizations, such as spectrograms and frequency graphs, should be displayed to the user within 5 seconds after the analysis is completed. This timely feedback allows users to interpret results effectively and make informed decisions.

By meeting these performance requirements, the vocal frequency recognition system will provide users with a robust and efficient tool for analyzing the vocal behaviors of the Jacobin cuckoo, ultimately supporting conservation efforts and ecological research

4.2 Security requirements

The security of the vocal frequency recognition system for the Jacobin cuckoo is paramount due to the sensitivity of the data involved, including audio recordings and any associated metadata. To safeguard this data, several security measures must be implemented:

1. **Data Encryption:** All audio recordings and associated metadata should be encrypted both in transit and at rest. This ensures that even if unauthorized access occurs, the data remains unintelligible without the proper decryption keys. Implementing protocols like HTTPS for data transmission and AES (Advanced Encryption Standard) for data storage will enhance security.

2. **User Authentication:** The system should require robust user authentication mechanisms to prevent unauthorized access. This can include multi-factor authentication (MFA), where users must verify their identity through multiple means (e.g., password and mobile verification). This measure ensures that only authorized personnel can access sensitive data and functionalities.

3. **Access Control:** A role-based access control (RBAC) system should be implemented to restrict user permissions based on their roles. For example, researchers may have access to data analysis tools, while administrative users may have full access to system settings. This limits exposure of sensitive functionalities to unauthorized users.

4. **Data Backup and Recovery:** Regular data backups are essential for recovering from data loss or breaches. The system should include automated backup procedures to

secure audio recordings and analysis results. A clear data recovery plan should be established to restore functionality quickly in case of an incident.

5. **Security Audits and Monitoring:** Regular security audits and monitoring of system logs should be conducted to identify and address potential vulnerabilities. Implementing intrusion detection systems can provide real-time alerts on suspicious activities, allowing for prompt intervention.

4.3 Software Quality Attributes

The success of the vocal frequency recognition system for the Jacobin cuckoo hinges on several key software quality attributes. Each attribute plays a critical role in ensuring that the system meets user expectations and performs reliably in real-world scenarios.

1. **Reliability:** The system must consistently provide accurate recognition and classification of the Jacobin cuckoo's vocalizations. Reliability can be ensured through extensive testing, including unit tests, integration tests, and system tests that mimic real-world conditions. The system should exhibit high uptime, with mechanisms in place for automatic recovery from failures.

2. **Maintainability:** The architecture should prioritize modular design, allowing for easy updates and enhancements. Comprehensive documentation, including code comments and user manuals, will facilitate future modifications. This is particularly important as new features may be added or existing functionalities improved based on user feedback and ongoing research.

3. **Usability:** The user interface must be intuitive and user-friendly, catering to a diverse audience, including researchers and conservationists. Usability testing should be conducted to gather feedback from actual users, ensuring that navigation is straightforward and that users can easily access the features they need. Providing clear instructions and support will enhance user satisfaction.

4. **Performance:** The system should process audio data and generate results efficiently, even with large datasets. Performance benchmarks, such as response times for audio recognition and analysis, should be established and monitored. Optimization techniques, such as using efficient algorithms and data structures, will help maintain performance levels.

5. **Scalability:** The software must be designed to accommodate growth, whether in the volume of audio data or the number of users accessing the system. This may involve implementing cloud-based solutions or scalable architecture that allows for increased computational resources as demand rises.

6. **Security:** Security measures should be woven into the system's fabric, ensuring that user data is protected from unauthorized access and breaches. Employing secure coding practices, regular security assessments, and adherence to industry standards will enhance overall software quality.

APPENDIX A: GLOSSARY

1. **SRS (Software Requirements Specification)**: A document that captures the functional and non-functional requirements for a system.
2. **Vocal Frequency Recognition**: The process of identifying bird species or other subjects based on their unique vocal frequencies.
3. **MFCC (Mel-frequency Cepstral Coefficients)**: A feature extraction technique used in audio processing, especially in speech and bird sound recognition, to represent the short-term power spectrum of sound.
4. **FFT (Fast Fourier Transform)**: A mathematical algorithm to compute the discrete Fourier transform (DFT) of a sequence, used in signal processing to convert signals from time to frequency domain.
5. **Spectrogram**: A visual representation of the spectrum of frequencies in a sound signal as they vary with time.
6. **Jacobin Cuckoo**: A migratory bird species known for its distinct vocalizations, used in this project for studying the relationship between its calls and weather changes.
7. **DBMS (Database Management System)**: Software for creating, managing, and interacting with databases to store and retrieve structured information.
8. **DSA (Data Structures and Algorithms)**: A fundamental set of concepts in computer science that is used to efficiently organize and manipulate data.
9. **Migration Patterns**: Refers to the habitual movement of species, like the Jacobin cuckoo, between regions, often influenced by environmental factors such as weather.
10. **Weather Data**: Meteorological information, such as temperature, precipitation, and humidity, used in this project to analyze correlations between climate conditions and the Jacobin cuckoo's vocal behavior.

APPENDIX B: ANALYSIS MODEL

Use Case Model:

The system's core functionality revolves around processing the audio files of Jacobin cuckoo calls and correlating this data with historical weather information. The key use case is "Vocal Frequency Analysis," which is broken down into sub-functions such as:

1. **Audio Input**: Uploading and reading WAV files.
2. **Feature Extraction**: Using FFT and MFCC to extract meaningful features from the bird calls.
3. **Data Correlation**: Analyzing historical weather data alongside extracted features to find patterns.
4. **Visualization**: Displaying spectrograms and graphs to present the findings.

Class Diagram:

The class structure includes:

1. **Main**: The entry point of the system where all functionalities are initiated.
2. **FeatureExtractor**: Extracts the MFCC, chroma features, and other relevant data from the audio signal.
3. **WeatherAnalyzer**: Correlates weather data with bird vocalizations.
4. **SpectrogramVisualizer**: Responsible for generating and displaying the spectrograms.
5. **DatabaseManager**: Manages the database interactions related to weather and audio files.

Data Flow Diagram:

The data flow within the system starts with the input of audio and weather data, passing through feature extraction and analysis processes, leading to outputs like correlation reports and visualizations.

APPENDIX C: ISSUES LIST

1. **Audio Data Quality**:

   Open Issue: Variability in the quality of audio files might affect the accuracy of vocal frequency recognition. Some files may have noise, low volume, or missing portions, which could distort feature extraction results.
   Resolution Strategy: Implementing noise reduction algorithms or discarding low-quality files.

2. **Inconsistent Weather Data**:

   Open Issue: There are gaps or inconsistencies in the weather data collected between 2000-2024, which may hinder effective correlation analysis.
   Resolution Strategy: Interpolating missing data points or focusing only on the available consistent data.

3. **Feature Extraction Accuracy**:

   Open Issue: Extracting meaningful features from bird calls, such as MFCC and pitch, can vary due to background noise or overlapping sounds.
   Resolution Strategy: Fine-tuning the feature extraction parameters (e.g., frame size) and exploring different libraries for accuracy.

4. **Data Integration**:

   Open Issue: Integrating the historical weather data with the extracted audio features to produce meaningful results is complex and might introduce data mismatches.
   Resolution Strategy: Developing a clear data schema and implementing validation checks during data integration.

5. **Real-time Processing Capability**:

   Open Issue: If the system is to handle real-time audio input in future versions, current algorithms (such as FFT) may need optimization for speed and efficiency.