BTP - 1 Report - 1

Data Collection Application

Under the guidance of Dr. Anil Kumar Vuppala, Speech Processing Lab

Neeruddula Veda Nivas Chowdary - 2021101037

Vempati Siva Koti Reddy - 2021101135

Introduction To The Application

Overview

The application in focus is a specialized web-based platform designed to facilitate efficient and interactive audio data collection linked to textual content. Primarily serving as a tool for audio annotation, the application caters to two distinct user roles: -administrators (admins) and normal users, each with unique functionalities and access levels. The core functionality revolves around users recording audio responses to text files provided within the application.

Purpose and Functionality

Admin Users:

- **Text File Management:** Admin users have the capability to upload and manage text files. This includes creating new annotations and overseeing the collection of audio data corresponding to these texts.
- **Annotation Overview:** Admins can view all existing annotations, monitor user contributions, and manage the overall content within the application.

Normal Users:

- **Audio Recording:** Normal users interact with the application primarily through recording audio. They are presented with text files and are required to record their audio responses to these texts.
- **Submission of Audio Data:** Once the audio is recorded, users can submit their contributions, which are then stored and managed within the application's database.

Objective

The aim of this application is to provide a streamlined, secure, and user-friendly platform for audio data collection and management. By addressing the specific needs of both admin and normal users, the application seeks to enhance the efficiency of audio annotation processes, ensure data security, and offer a cohesive and integrated user experience.

Issues Observed In The Existing Application

1. Inefficiencies and Limitations in Database Management

- **Audio File Update Problems:** Inconsistencies and failures in updating audio files to the MongoDB database, impacting data integrity and reliability.
- **Dependence on Google Drive for Audio Storage:** Currently, audio files are initially stored in Google Drive, and then the links are stored in MongoDB. This process is cumbersome and less efficient than using dedicated object storage solutions like MinIO.

2. User Interface (UI) Malfunctions

- **Non-Functional Audio Recording Button:** The audio recording feature is non-operational. Users are unable to record audio corresponding to the provided text, severely hindering the core functionality of the application.

3. Lack of Data Security

- **Unrestricted Data Access:** All users currently have the ability to access and alter metadata, posing significant data security and integrity risks.

4. Fragmented Application Structure

- **Multiple Disjointed Applications:** The presence of three different applications for related functionalities leads to inefficiencies and user inconvenience. The goal is to integrate these into a single, more efficient, and reliable application.

5. Absence of User Authentication System

- **No Login/Register Feature:** The application lacks a mechanism to store user data securely and to provide role-based access functionalities (admin/normal users), limiting user management and personalization capabilities.

Our Objectives And Goals

Overview

In our endeavor to enhance our web-based audio data collection and management platform, we have identified key objectives and goals. These are centered on addressing current system limitations and leveraging advanced technology to offer a more effective, secure, and user-friendly solution.

Core Objectives

1. Comprehensive Redesign and Redevelopment

- **Objective:** Undertake a complete redevelopment of the existing application to ensure a more robust, scalable, and efficient system.
- **Goal:** To provide an improved, intuitive interface and backend, enhancing overall performance and user interaction.

2. Secure User Authentication System

- **Objective:** Implement a SignIn/SignUp feature for admin and normal users.
- **Goal:** Facilitate role-specific access to functionalities, improving security and delivering a tailored user experience. Admins will have advanced capabilities like metadata management, while normal users will focus on audio-related tasks.

3. Enhanced Admin Functionalities

- **Objective:** Provide admins with the tools to upload, view, and manage metadata and corresponding audio files.
- **Goal:** Streamline content management and administrative oversight within the application.

4. Integration with MinIO for Audio Handling

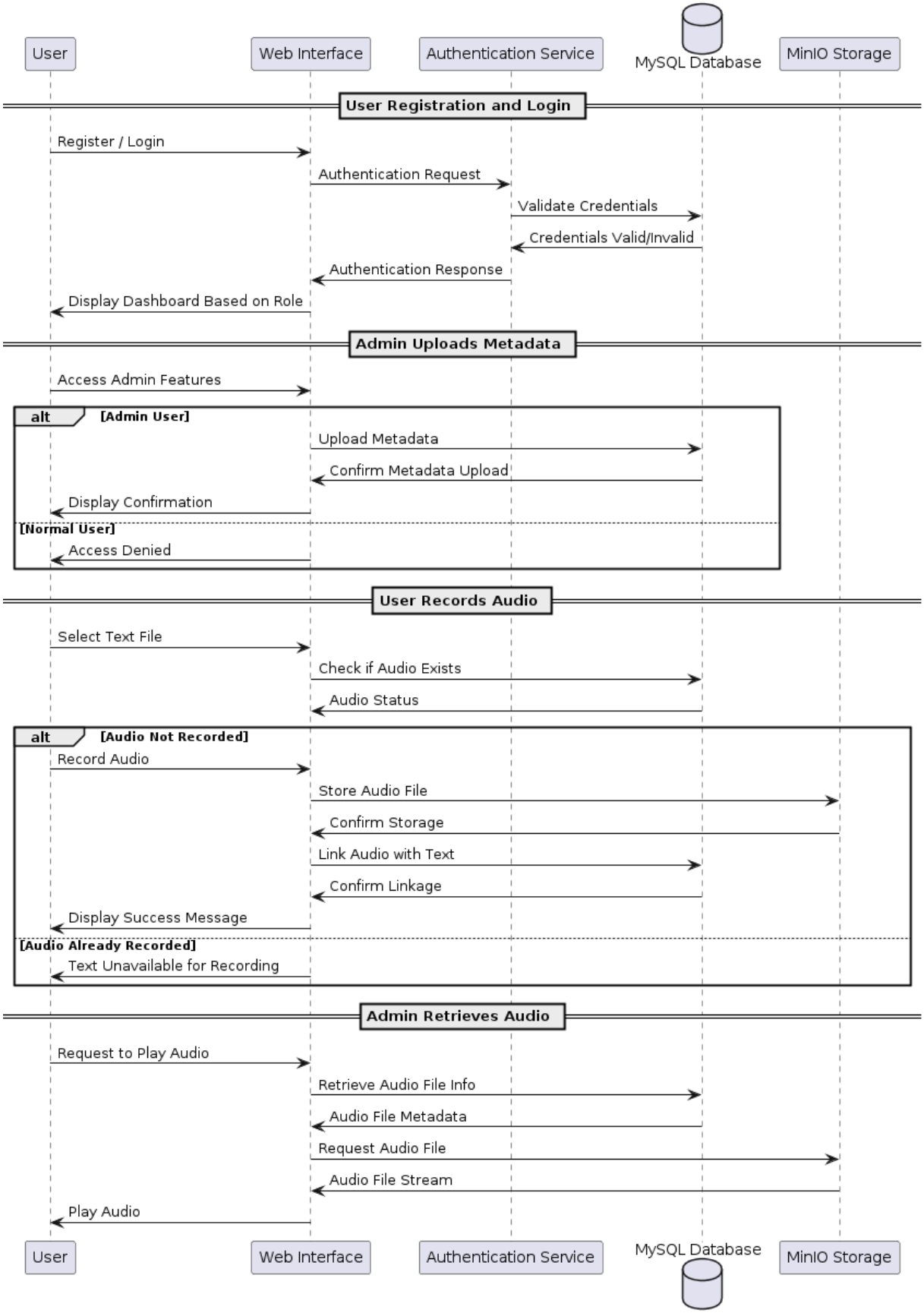
- **Objective:** Utilize MinIO for secure and efficient audio file storage and retrieval.
- **Goal:** Enable straightforward audio playback within the application, offering users the ability to listen to submitted audio files with a simple click.

5. Robust Database Management and Linking

- **Objective:** Employ MySQL for metadata storage and create a relational link between MinIO audio files and MySQL metadata.
- **Goal:** Establish a dependable system linking audio files to their metadata, ensuring data integrity and accessibility. This involves uniquely indexing audio files in MinIO to correspond with MySQL keys.

6. Dynamic Display of Text Files for Recording

- **Objective:** Implement a system to display text files to users that have not yet been associated with an audio recording.
- **Goal:** Once a user submits an audio recording for a text file, that text file will no longer be available for recording to other users. This ensures each text file receives a unique audio response and prevents duplication of effort.


User
Web Interface
Authentication Service
MySQL Database
MinIO Storage
User Registration and Login
Register / Login
Authentication Request
Validate Credentials
Credentials Valid/Invalid
Authentication Response
Display Dashboard Based on Role
Admin Uploads Metadata
Access Admin Features
alt
[Admin User]
Upload Metadata
Confirm Metadata Upload
Display Confirmation
[Normal User]
Access Denied
User Records Audio
Select Text File
Check if Audio Exists
Audio Status
alt
[Audio Not Recorded]
Record Audio
Store Audio File
Confirm Storage
Link Audio with Text
Confirm Linkage
Display Success Message
[Audio Already Recorded]
Text Unavailable for Recording
Admin Retrieves Audio
Request to Play Audio
Retrieve Audio File Info
Audio File Metadata
Request Audio File
Audio File Stream
Play Audio
User
Web Interface
Authentication Service
MySQL Database
MinIO Storage

Methodology and Tech Stack for Application Redevelopment

Development Methodology

Our approach to developing the application will be iterative and agile, ensuring flexibility, continuous improvement, and incorporating feedback at every stage. Key phases of this methodology include:

1. Requirements Gathering: In-depth analysis of the existing application to understand the functional and technical requirements.

2. Design and Prototyping: Creating initial designs and prototypes for the application interface and architecture.

3. Development and Iteration: Coding the application in phases, allowing for regular review and adjustment.

4. Testing: Rigorous testing to ensure functionality, usability, and security.

5. Deployment and Monitoring: Launching the application followed by continuous monitoring and maintenance.

Technology Stack

Frontend Development

React: We will utilize React for the frontend development. This JavaScript library is renowned for its efficiency and flexibility in building interactive user interfaces. It will enable us to create a responsive and dynamic front-end experience for the application.

Backend Development

Express and Node.js: For the backend, we will employ Express.js, a web application framework for Node.js. This combination allows for building fast, scalable, and efficient server-side applications. Node.js's non-blocking I/O model ensures efficient processing and handling of multiple requests simultaneously.

Database Management

MySQL: The metadata, including the text of the audio, and the unique identifiers (IDs) for the audio files, will be stored in a MySQL database. MySQL is chosen for its reliability, performance, and support for complex queries, which is crucial for efficient data management and retrieval.

Audio File Storage

MinIO: Audio files will be stored in MinIO, an object storage solution known for its high performance and scalability. MinIO's compatibility with cloud-native environments makes it an ideal choice for storing large amounts of unstructured data like audio (wave) files. The use of MinIO will facilitate efficient storage and retrieval of audio content, and its unique naming convention will aid in the easy mapping of audio files to their corresponding metadata in MySQL.