

Problem statement.

The Trentoniana archive of audio recordings is disorganized, presenting an overwhelming amount of information to the average user that they will have no understanding how to navigate. Limited or incorrect filtering options (two of the five provided filters are “New Jersey”), limited synopses of the recordings, and general information overload are some of the issues with the current way they are presented. In addition, with both audio recordings and their respective transcripts being located on completely different websites than Trentoniana’s, it is difficult to listen and view both at the same time for comparison and transcript revision purposes. The creation of an all-in-one interface for the recordings and transcripts on the Trentoniana website would be preferable. This would be an interface that allows a user to freely navigate between oral histories by applying meaningful parameters (such as filtering by topic or date) to accurately find what they are looking for.

Objective of the module.

The objective of our module is to provide an easy-to-use interface accessible to users who may not have the highest proficiency in technology-usage to view and filter through Trenton residents’ oral histories and their transcripts simultaneously. In other words, our module’s objective is to create an alternate, improved version of the archive.org website for the Trentoniana audio collections.

Description of the desired end product, and the part you will develop for this class.

Our team’s development goal is to group Trenton residents’ oral histories (audio files) and their transcripts together onto the same webpage for easy comprehension of said audio. We will develop a database using PostgreSQL that allows the average user to easily filter these audio files based on parameters they provide (e.g. date, location, topic, date added, etc.), in addition to a clean interface that allows a user hear a recording and see its respective transcript at the same time. In terms of user access, privileged users/administrators should have the ability to edit and revise transcripts at will for better workflow. If they find an error, unprivileged users should also be able to suggest edits by commenting on transcripts and have those edits be officially added upon administrator approval.

Description of the importance and need for the module, and how it addresses the problem.

The Trentoniana Collection aims to preserve a local history pertaining to the lives of Trenton, New Jersey residents. Their pursuit aims to enrich the quality of life in the City and pass on the stories of its people's heritage and diversity. At its current state, the archive leaves much room for improvement in its organization and its ease-of-access to the public. All current oral histories of Trenton residents are located on a third-party website that presents the average user an overwhelming amount of data that they will have no idea how to traverse. Our module aims to address these ambiguities with the current system by presenting an easily accessible interface for Trenton residents' oral histories that is sortable through a variety of different filters pertaining to user specifications. For example, the user should be able to filter oral histories by their date and topic of conversation. In addition, a transcript will have the option to be provided with each audio file to keep resident accounts understandable.

Plan for how you will research the problem domain and obtain the data needed.

- Download audio files from archive.org
- Collect transcripts from associated data storage

Other similar systems / approaches that exist, and how your module is different or will add to the existing system.

Phonemica (<https://phonemica.net/>) is an audio collection of Chinese topolects and dialects for archival and preservation purposes. This is not open source and is highly specialized. Instead, we will build a more generic system.

Possible other applications of the system (how it could be modified and reused.)

Our database has the potential to also be used for other general-purpose audio database applications, such as a database that holds audio for songs (instead of Trenton resident oral histories) and their lyrics (transcripts). A variety of filters and parameters should be able to be specified by the user in order to find the song they're looking for. For example, if they only know the year and the artist of a song, they should be able to search for the song using these filters.

Performance – specify how and to what extent you will address this.

- Writing good database queries
- Using database indexes intelligently
- Hosting audio files and transcripts directly on our own server

Security – specify how and to what extent you will provide security features.

Security features of our module include the prevention of unauthorized editing of transcripts or any other data, in addition to providing administrative roles for authorized users (i.e. users who are allowed to edit the database).

Backup and Recovery – specify how and to what extent you will implement this.

- Nightly dumps of audio data to an S3 server
- Nightly incremental backups of the database

Technologies

- Programming Language (e.g. Rust, PHP, C#, etc.)
- Backend Web Framework (e.g. warp/thruster, ASP.NET Core)
- Frontend Web Frameworks (e.g. React/Vue/vanilla JavaScript)

Database Concepts

- PostgreSQL/Relational Databases
- Database modelling and design
- Database security
- SQL queries
- UML in data modelling

A diagrammatic representation of the system boundary that specifies what data you will model and which queries you will implement.

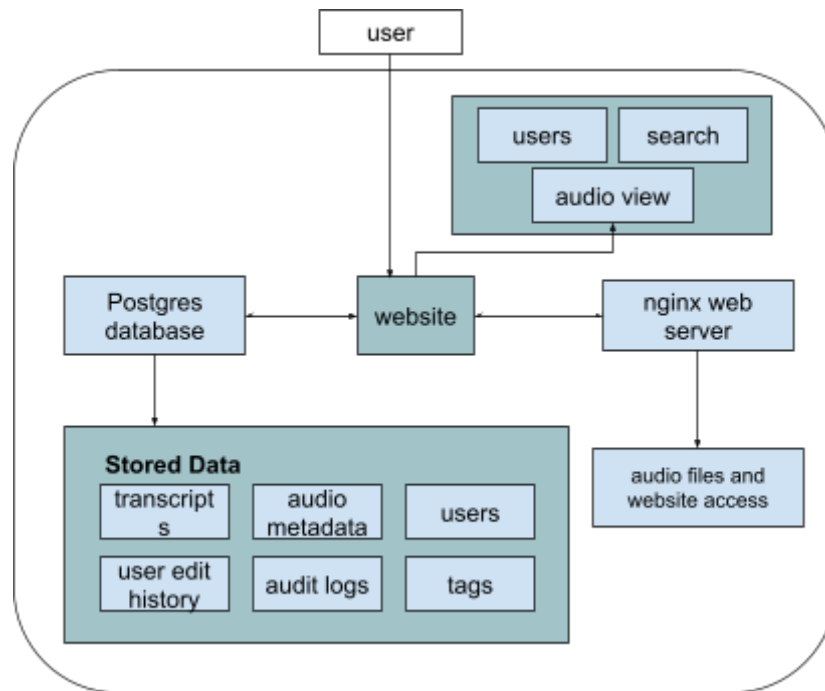


Figure 1. System Boundary Diagram



CSC 315 Project

Alexander Benasutti, Raymond Chow, Sam Haseloff, Alexander Viola

Need

- A general portal for Trentonian oral history archives which is accessible to users of all technological proficiency levels
- Better searching options, instead of “N.J.” and “New Jersey” (all of the audio are from NJ by definition, so it is redundant)
- Audio and transcript or summary view

Approach

- Postgres database to store data
- Website to simultaneously listen to an audio file and read a transcript
- Easy-to-use, flexible administration tools
- Ability to search and sort through transcripts for users
 - Prevent information overload

Benefit

- Ease of access (UX) for users and contributors
- Synopsis of audio recordings helps users select what’s most relevant
- A system that could be considered first-party, instead of being on archive.org
- Better accessibility for consumers (A11Y)
- Ability to host files in one centralized, organized location

Competition

Other approaches

- Often closed-source
- Heavy-weight, stored with other users’ data
- Highly specific

Our approach

- Open source
- Lightweight and cheap to host
- More generalized

Figure 2. 1-Page Quad Chart