Cover Page

Team Number 5
Trentoniana Transcription & Database Implementation

- Nina Brossa - brossan1@tcnj.edu
- Sarah Hoffman - hoffms11@tcnj.edu
- Dean Peppe - pepped1@tcnj.edu
- Alyssa Popper - poppera1@tcnj.edu

Final Project Report

Executive Summary

While the existing website for the Trentoniana oral history website isn't terrible, there are certainly features that could be improved. Specifically, our group thought the user interface (UI) and website navigation could use some work, as well as the addition of transcripts for each entry. For each of these discussions, we were sure to include our stakeholder and took their feedback heavily into consideration

Our approach involved creating a new website for the Trentoniana archives, including a database for the archives created with PostgreSQL and user-friendly web pages written with Flask and HTML. For the database, the first thing we wanted was to make sure it had all the features of the old website. This includes multiple searchable keywords and participants per archive, archive descriptions, interview date, and audio files. We also added a section for places, in case the Trentoniana ever expands their oral histories beyond Trenton, and the transcripts. Additionally, we also wanted to make the database scalable to allow users to add features in the future, so we used good database creation habits and formatting. For our front end work, we had a heavy focus on a good UI and easy website navigation. On our Search page, we didn't want to overwhelm the user, so we have limited searching features. Like the database features, we wanted to keep some of the UI features in the old website too, such as the ability to play the oral history files from the page.

The benefits to our approach are clear compared to the older website. When searching for archives, the user has many options, but not an overwhelming amount. While our search results might display just the title and interview date, our search function searches deeper into the database behind the scenes. It's also very easy to include one or multiple keyword filters in a search using the check boxes on the side. On each Show More page, there is now a link to the corresponding transcript, written by our stakeholders. With how our database is constructed, it should also be very easy to create, edit, or delete entries, keywords, participants, etc. from the website itself.

To use our project as a direct replacement to the old website, there would be several costs to consider. The first major cost would be completing our project. In its current state, many of its features are incomplete. The account functions, the table operations, and keyword functions are all incomplete, so there would need to be time, effort, and potentially money spent trying to fix them. The second major cost would be running the project and everything that comes with that. If the website and database would be run on local hardware, you would have to consider having separate hardware for the project and security features to prevent people from accessing it. If you were to have the project hosted from a cloud, then you have to worry about the monthly fee instead of hardware.

Elaboration: Project Proposal and Specifications

Stage II:

Problem:

The Trentoniana Library is a database that stores information related to the time where, after World War II, many migrated out of Trenton. It is important that anyone who wants to access this information can do so. However, the way the website currently presents its audio recordings is not too user-friendly. Audio clips do not have transcripts, are presented with non-uniform images of the recordings' sound waves, and are sorted by the number of views a recording has by default. If the database is difficult to navigate, it prevents people from being able to properly research what they need to.

Objective:

The objective of the module we are proposing is to make the information in the Trentoniana Library's database more accessible, and to fix the problems the website currently has in order to make it more user friendly. Fixing the sorting method will make it easier to search for audio files, and fixing how the audio clips are presented will make it easier to retrieve them from the database. We would also like to add transcripts of the audio recordings. It is important that those hard of hearing, or those who learn better visually, can access the material and learn as much as they can from it. This way, the information can still be retrieved. We believe taking these steps will make the database easier to navigate.

Desired End Product:

We would simply like to edit the web page, or create a site that is similar but more accessible. We would remove the "sort by views" option, insert the transcript, and sort only in list form. This is what we plan on developing in class.

Importance of the Module:

It is important that our module provides more accessibility than the current website does. First, most people do not search for works by number of views when conducting research. They mainly search by date, author, and topic. Therefore, sorting the audio recordings by views, especially by default, just makes things confusing. Eliminating this method of sorting would make searching easier and provide more relevant works for the user. Second, without transcripts of the audio recordings, people who are hard of hearing or who learn better visually cannot access the information. It's important for us to add the transcripts so that everyone has access. Third, the tab view of the audio files is disorienting; the soundwave images are not uniform, which makes the jagged lines appear in all different places. There is a list view, but again, it is not the default. We would like to eliminate the tab view, to make the webpage easier on the eyes.

Plan for Researching the Problem Domain and Obtaining Data:

The research for the problem domain and the data will be provided by the Trentoniana Library (https://trentonlib.org/trentoniana/audio-visual/) in addition to the class. Transcripts of the audio recordings will be provided by a transcriber. Some outside sources like search engines might also be used at some points.

Other Similar Systems and How Our System is Different

A system that is similar to ours is The North Carolina Folklife Institute's institute. This system stores many different things, from artists to publications, while our system will just store audio files. Our system will have a more modern user interface and will also have searching and sorting capabilities. Another system that is similar to ours is Schomburg Center Research/data. This system allows you to search online books, journals and more. It works by searching by title, subject, key word, and ISBN. Our system will just store audio files and will have similar search criteria such as title and subject.

Other Applications for the System:

The system could be revised to be reused in slightly different capacities for the Trentoniana Library. For example, it could be relatively the same system with similar search queries and the much better user interface, however it can hold different information other than audio files, such as video files and more. These changes would only be miniscule for the system and the Trentoniana Library would be able to have a standard system for all of their files with some slight modifications to the audio file system.

Performance:

We will try our best to optimize our algorithms to search the database. We will also design an elegant yet simple web interface to ensure that most of the computing time can be reserved for the back end database operations.

Security:

In order to address security concerns, we will need to have certain permissions given to certain users. The exact structure of these roles and permissions is subject to change based on the needs of the customer and the structure of the staff at the Trentoniana Library, but the basic concept will include an admin, editor, and user account, with possible intermediate accounts with other permissions. The admin account will have the ability to add, remove and edit entries and permissions from other accounts, as well at the permissions that the other accounts have. The editor account can add, remove and edit entries, but cannot change permissions or perform potentially fatal actions such as delete the entire database. The user account would be read only. At the request of the client other accounts could be added with varying permissions.

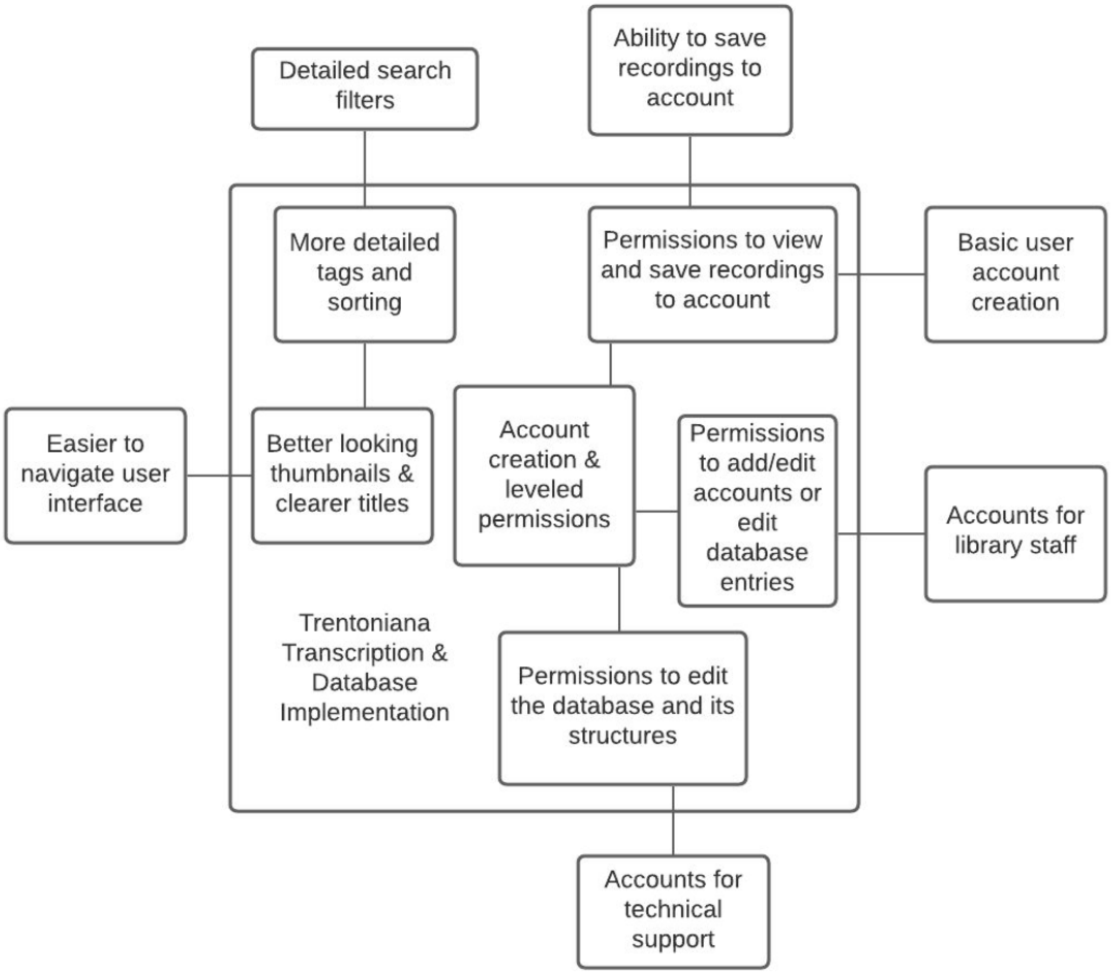
Backup and recovery:

Recovery must be possible in the event of a failure. In order to allow recovery in almost every situation (with the exception of a disk failure or catastrophic failure), we will have a system log, an append only file that will contain all previous commands given to the database. This way we will be able to recover the database in the event of failure.

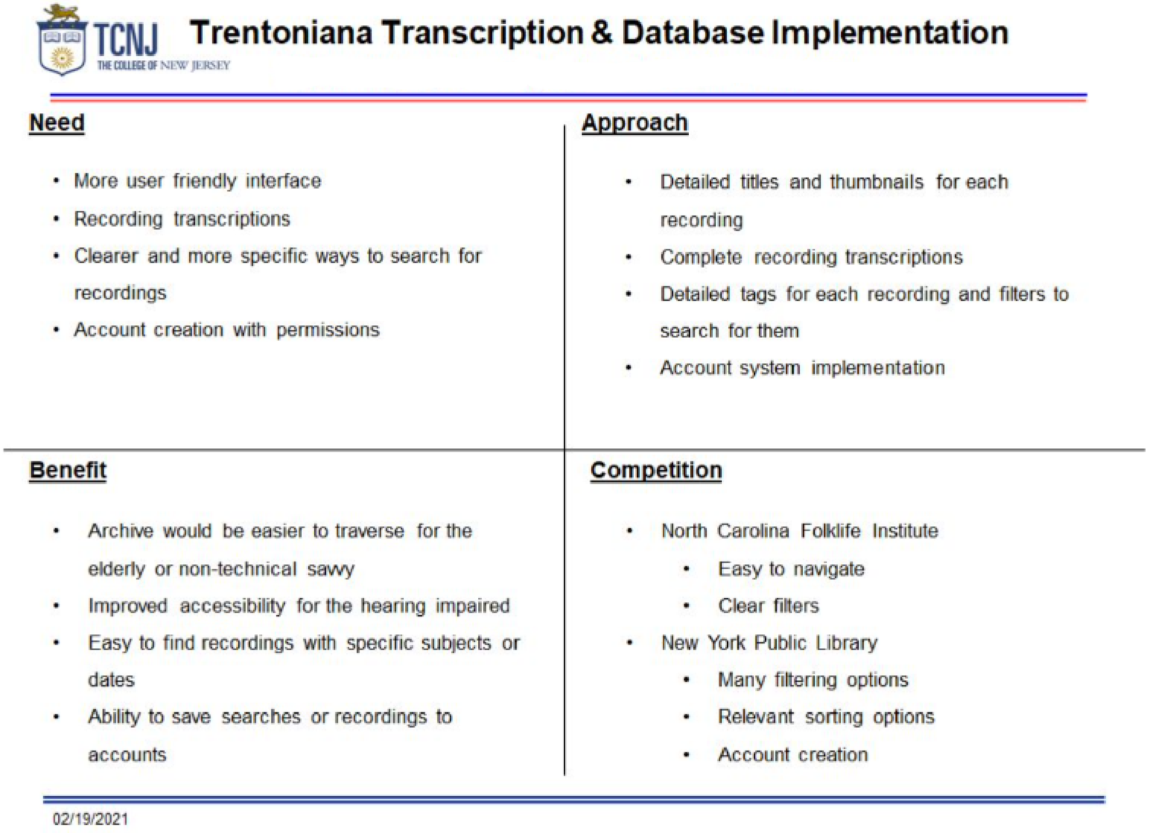
Technologies We Need to Learn:

As a group, we will need to learn and understand a couple different technologies to effectively work on this project. The back-end workers will need to understand SQL for implementing the database and GitHub for proper version control. Our transcriber will have to learn how to effectively use the audio transcription software.

System Boundary Diagram:


Detailed search filters
Ability to save recordings to account
More detailed tags and sorting
Permissions to view and save recordings to account
Basic user account creation
Easier to navigate user interface
Better looking thumbnails & clearer titles
Account creation & leveled permissions
Permissions to add/edit accounts or edit database entries
Accounts for library staff
Trentoniana Transcription & Database Implementation
Permissions to edit the database and its structures
Accounts for technical support

Quad Chart:



TCNJ THE COLLEGE OF NEW JERSEY

Trentoniana Transcription & Database Implementation

Need

- More user friendly interface
- Recording transcriptions
- Clearer and more specific ways to search for recordings
- Account creation with permissions

Approach

- Detailed titles and thumbnails for each recording
- Complete recording transcriptions
- Detailed tags for each recording and filters to search for them
- Account system implementation

Benefit

- Archive would be easier to traverse for the elderly or non-technical savvy
- Improved accessibility for the hearing impaired
- Easy to find recordings with specific subjects or dates
- Ability to save searches or recordings to accounts

Competition

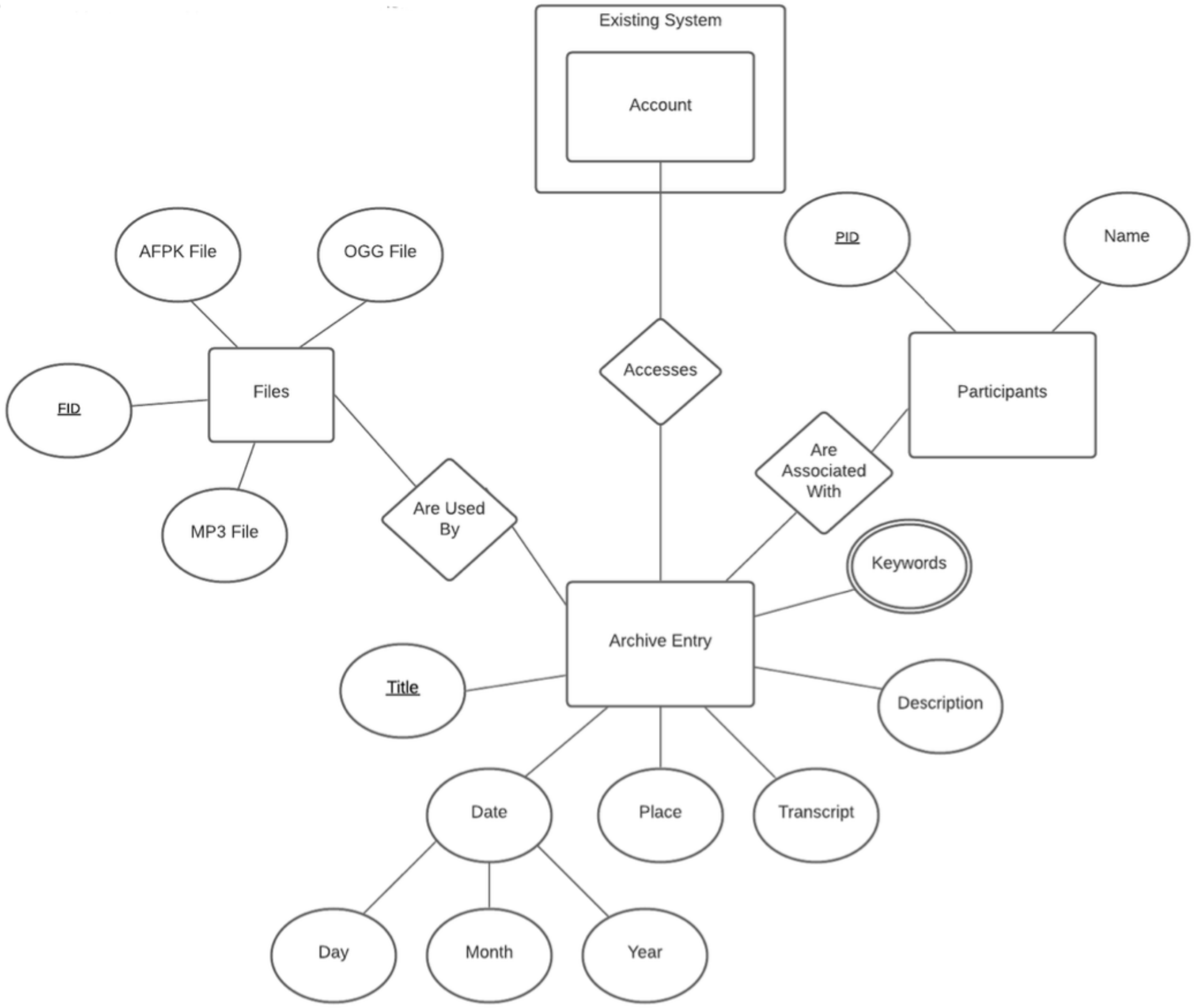
- North Carolina Folklife Institute
  - Easy to navigate
  - Clear filters
- New York Public Library
  - Many filtering options
  - Relevant sorting options
  - Account creation

02/19/2021

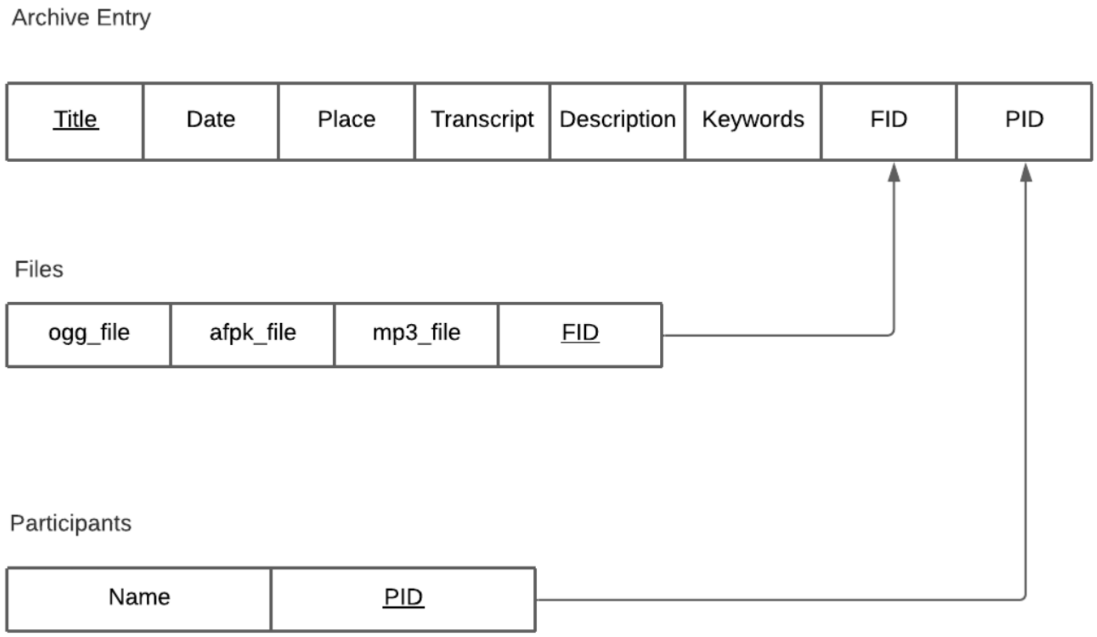
Elaboration: Design

Stage III:

Transcription and Database Implementation:



Database Schema:



Database Estimations:

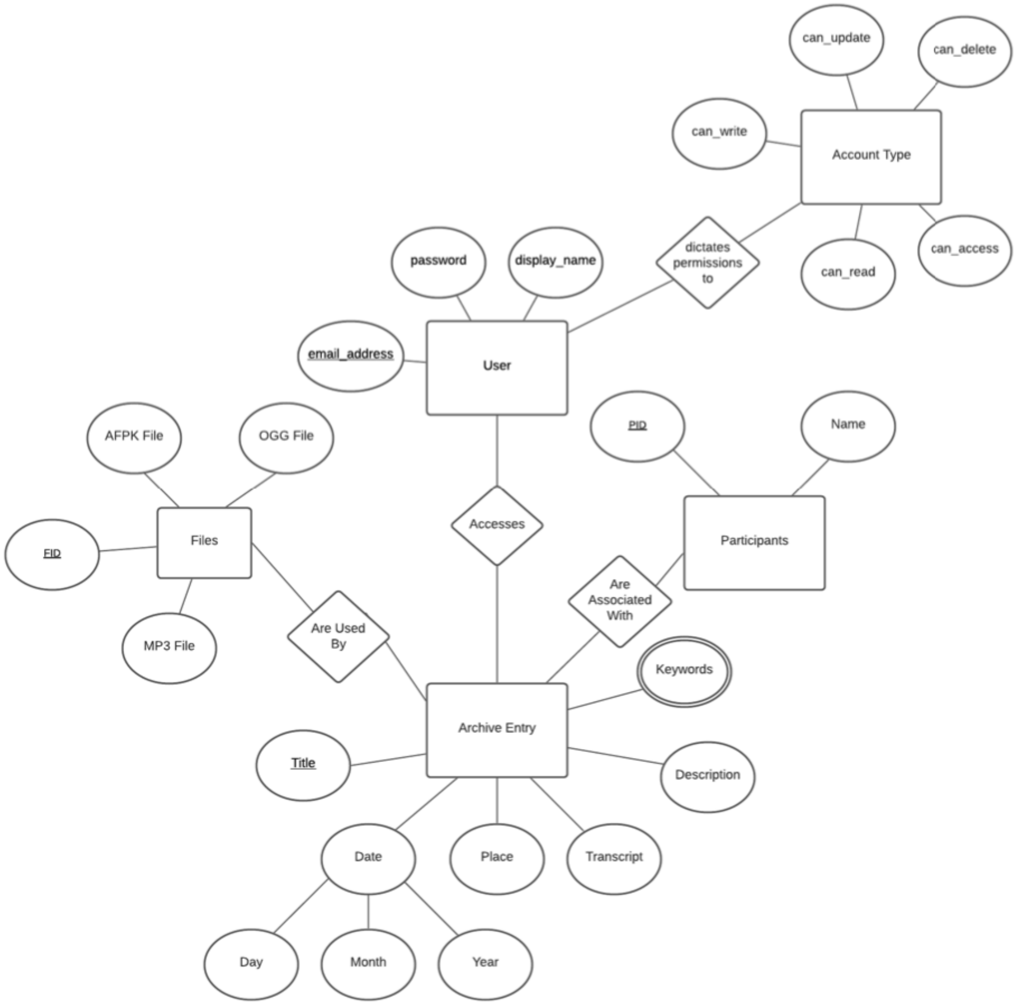
Estimated Database Size: 74 audio files and 74 Transcripts (assuming we get all of them done; if we don’t, probably around 30)

Estimated Average number of searches: 2,218 total views of all files/74= 30

Estimated Types of searches: 16 (by topic and date)

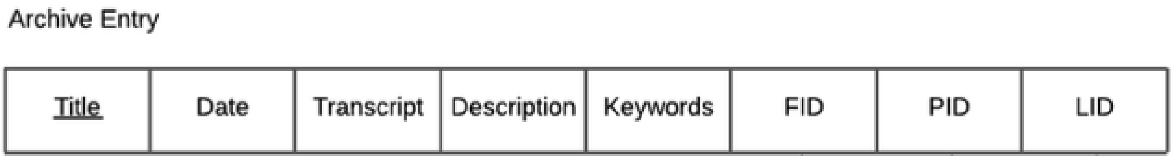
Stage IV:

Schema:



Relations and Their Forms:

1.

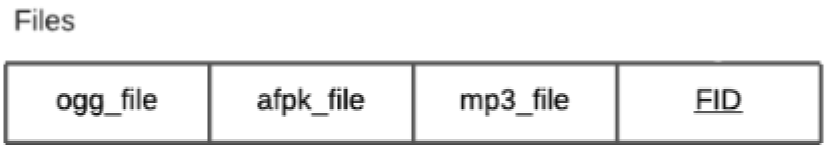


Archive Entry

Title	Date	Transcript	Description	Keywords	FID	PID	LID

The archive entry table is in BCNF because it is in 3NF and for every functional dependency Title -> {Date, Transcript, Description, Keywords}, Title is the super key of the table.

2.

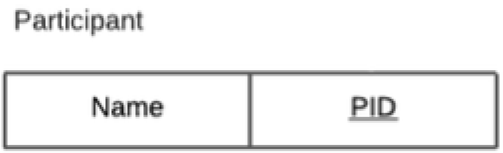


Files

ogg_file	afpk_file	mp3_file	FID

The files table is in BCNF because it is in 3NF and for every functional dependency FID -> {ogg_file, afpk_file, mp3_file}, FID is the super key of the table.

3.

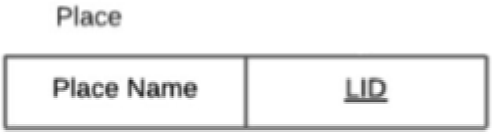


Participant

Name	PID

The participant table is in BCNF because it is in 3NF and for every functional dependency PID -> {Name}, PID is the super key of the table.

4.

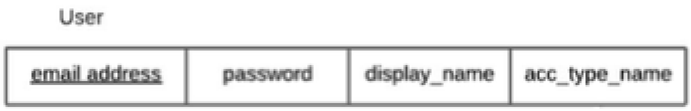


Place

Place Name	LID

The place table is in BCNF because it is in 3NF and for every functional dependency LID -> {Place Name}, LID is the super key of the table.

5.

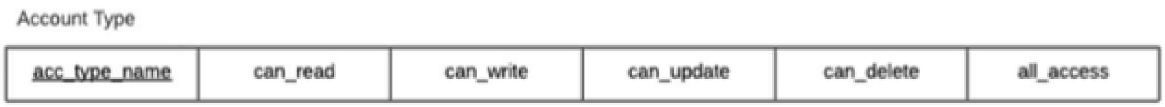


User

email address	password	display_name	acc_type_name

The user table is in BCNF because it is in 3NF and for every functional dependency email_address -> {password, display_name}, email_address is the super key of the table.

6.



Account Type

acc_type_name	can_read	can_write	can_update	can_delete	all_access

The account type table is in BCNF because it is in 3NF and for every functional dependency acc_type_name -> {can_read, can_write, can_update, can_delete, all_access}, acc_type_name is the super key of the table.

Virtual Tables:

1. SAME_PARTICIPANT:

   Join the Archive Entry table and the Participant table where the PIDs are equal. This will assist in retrieving archive entries by participant should the user search for a participant. It will also assist in the creation of other views.

2. MAIN_INDEX:

   Join SAME_PARTICIPANT with Place by LID. Select the Title, Date, Place Name, and Participant name. These will be the things that are displayed on the main entries page.

SQL Queries:

1. Return_Administrators

   SELECT display_name FROM User WHERE acc_type_name = “admin”;

2. Return_Archives_From_Trenton

   CREATE VIEW trentonLID AS
   SELECT LID FROM Place WHERE Place_Name = “Trenton, NJ”;
   SELECT * FROM Archive_Entry WHERE LID = trentonLID;

3. Create_Participant

   INSERT INTO Participant VALUES (‘James Wnorowski’, ‘0052’);

4. Create_New_Account
5. INSERT INTO User VALUES (‘pepperd1@tcnj.edu’, ‘password1’, ‘pepperd1’, ‘admin’);

6. Update_Password

   UPDATE User SET password = “SecurePassword2” WHERE email_address =“pepperd1@tcnj.edu”;

7. Return_Jewish_Archives

```
SELECT * FROM Archive_Entry WHERE Keywords = ‘Jewish’;
```

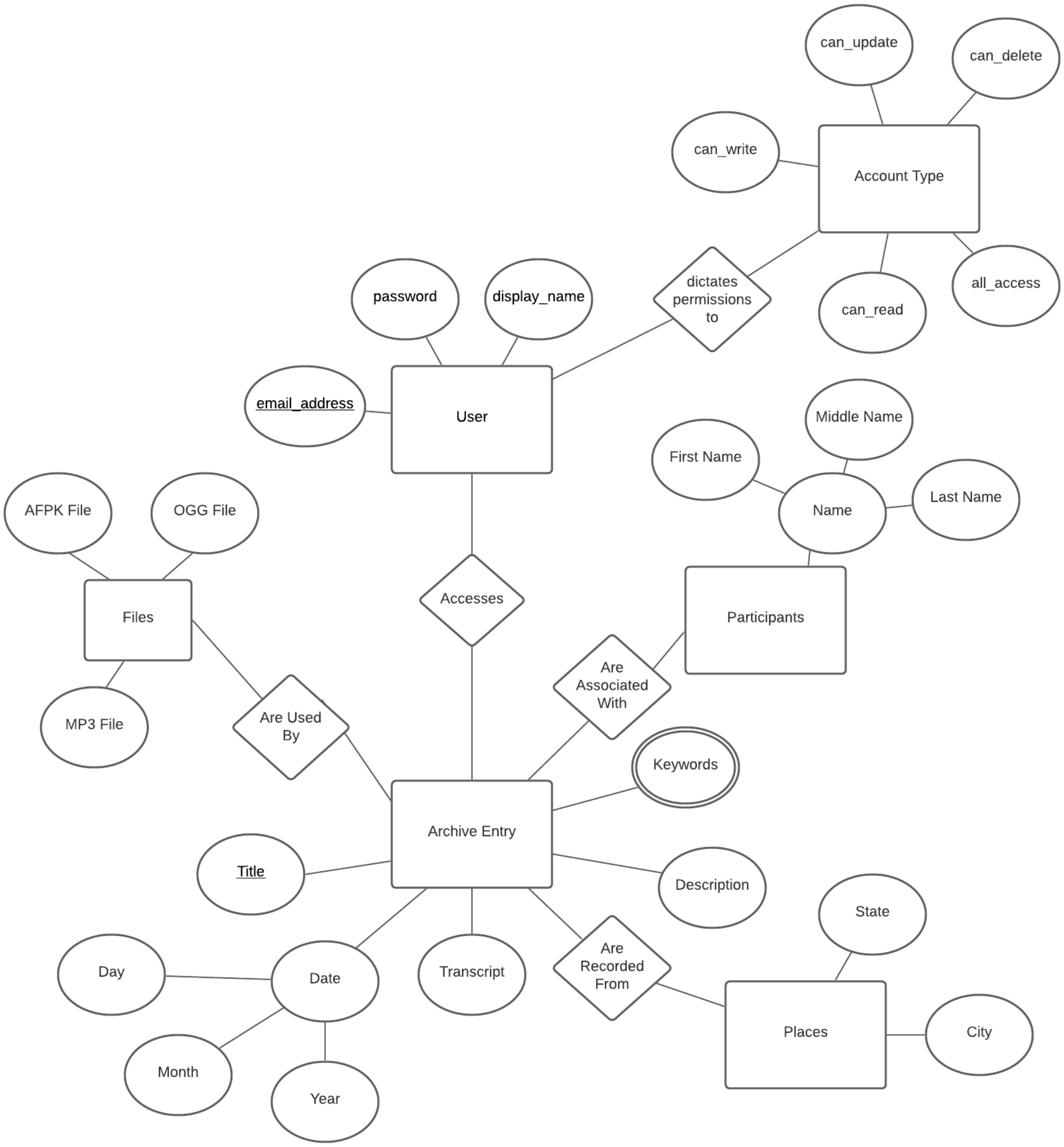
8. Sort_Archives_By_Oldest:

```
SELECT * FROM Archive_Entry ORDER BY Date ASC;
```

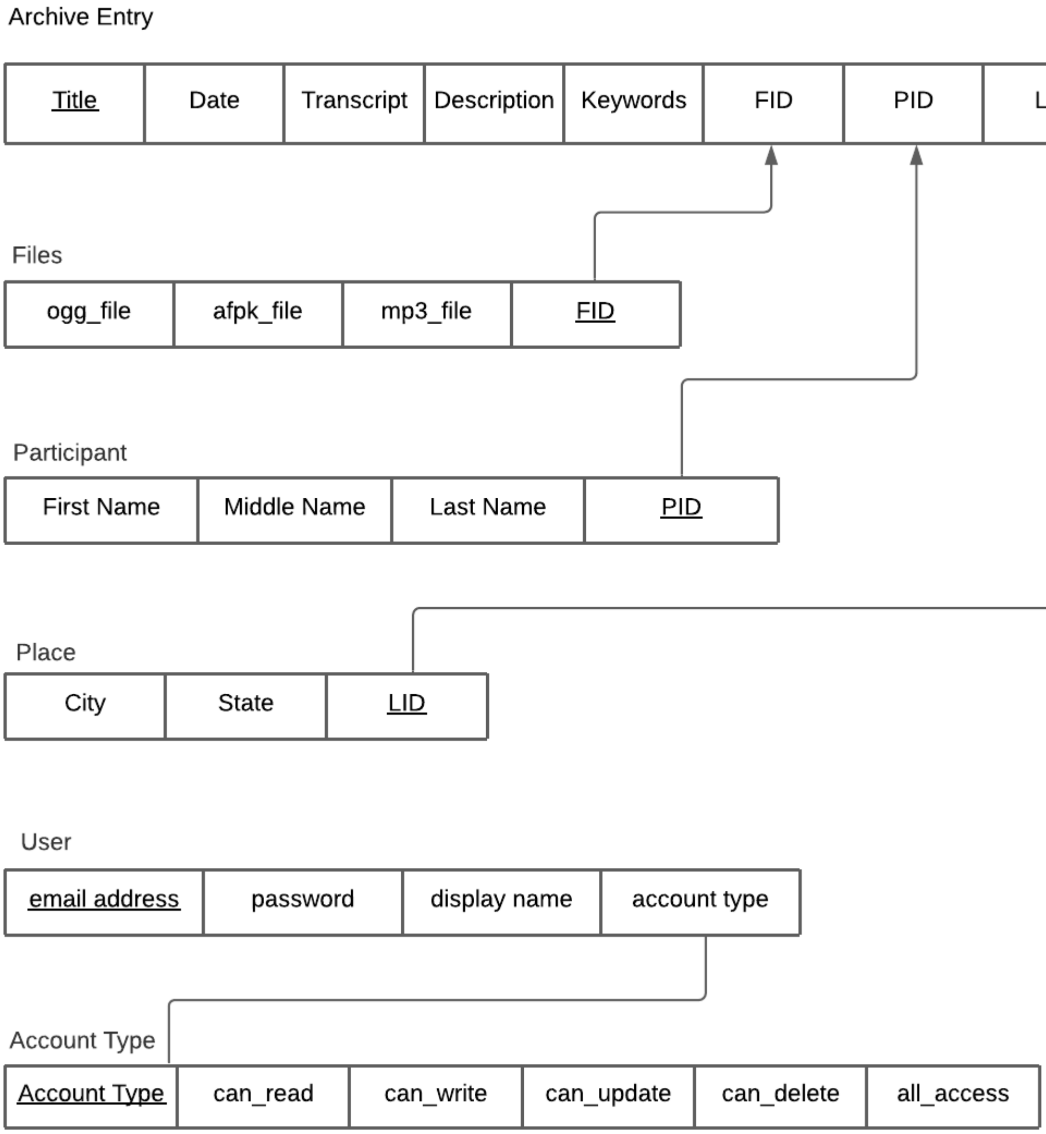
Elaboration: Tables, Queries, and User Interface:

Stage V (Original):

ER Diagram:



Schema:



Archive Entry

Title	Date	Transcript	Description	Keywords	FID	PID	LID

Files

ogg_file	afpk_file	mp3_file	FID

Participant

First Name	Middle Name	Last Name	PID

Place

City	State	LID

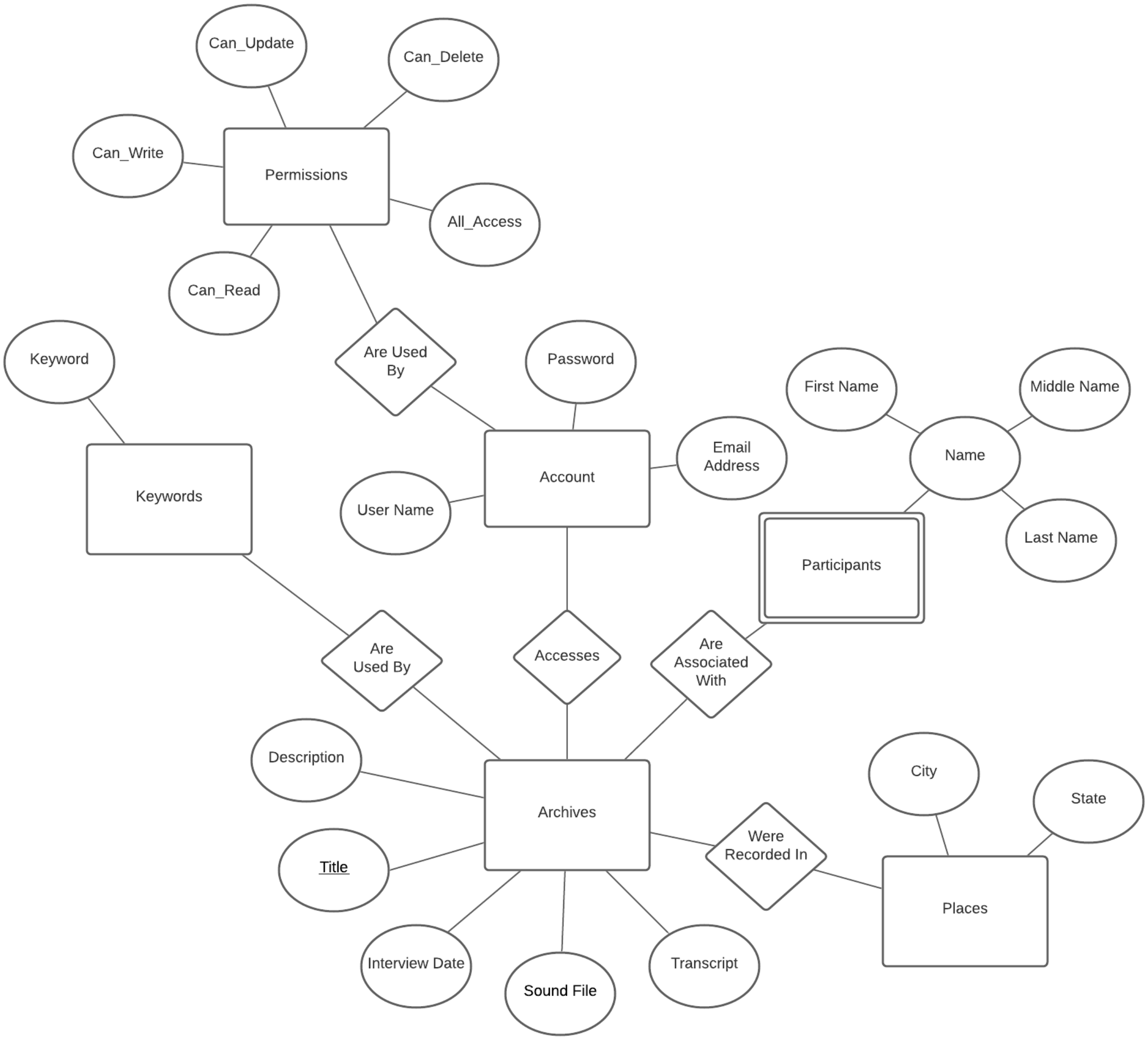
User

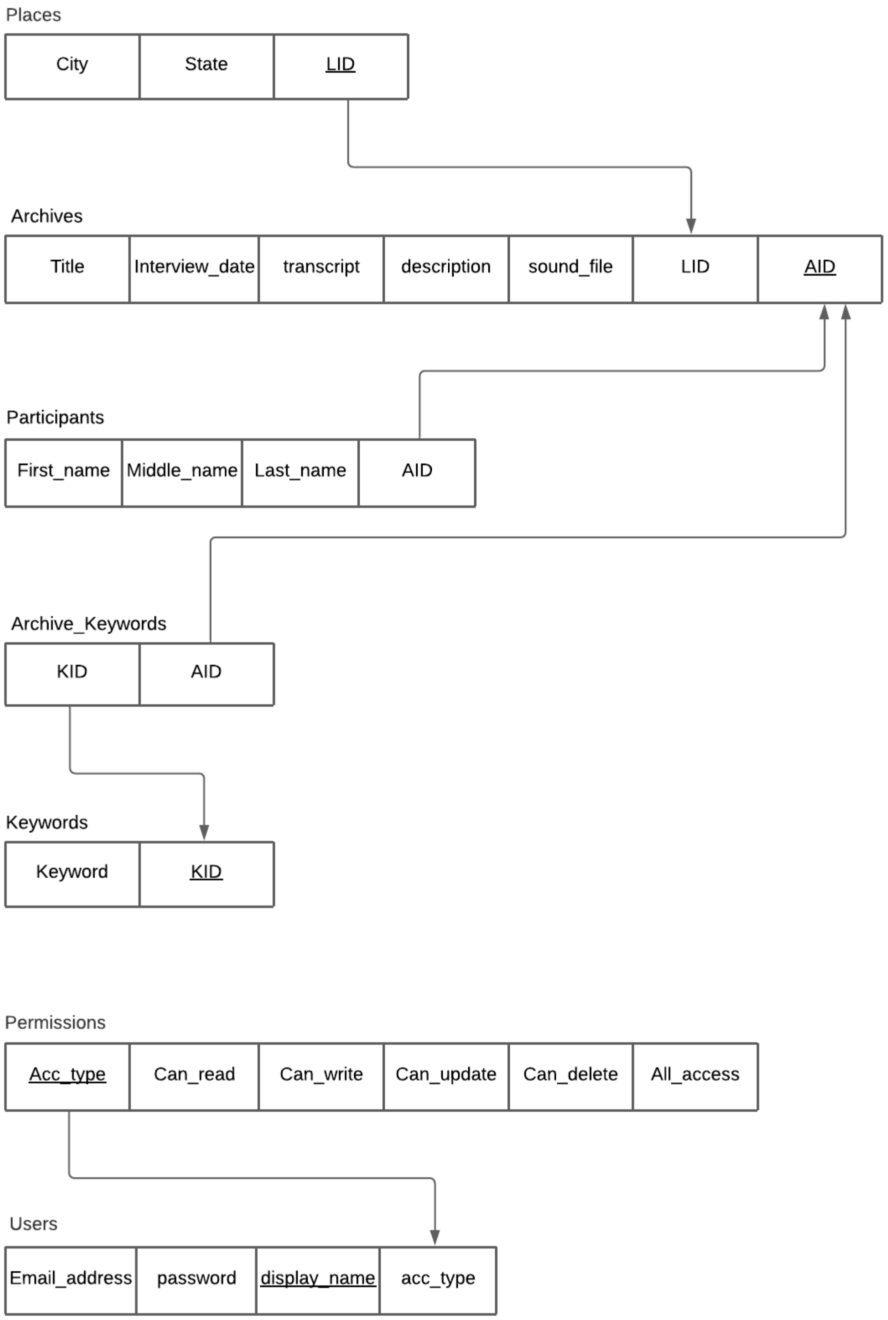
email address	password	display name	account type

Account Type

Account Type	can_read	can_write	can_update	can_delete	all_access

Stage V Revisions - Final Product:


Can_Update
Can_Delete
Can_Write
Permissions
All_Access
Can_Read
Keyword
Are Used By
Password
First Name
Middle Name
Name
Email Address
Account
User Name
Keywords
Last Name
Participants
Are Used By
Accesses
Are Associated With
Description
Archives
City
State
Were Recorded In
Title
Places
Interview Date
Sound File
Transcript



Places

City	State	LID

Archives

Title	Interview_date	transcript	description	sound_file	LID	AID

Participants

First_name	Middle_name	Last_name	AID

Archive_Keywords

KID	AID

Keywords

Keyword	KID

Permissions

Acc_type	Can_read	Can_write	Can_update	Can_delete	All_access

Users

Email_address	password	display_name	acc_type