Part B: Database Design

David Norton 3616902

Tye Shutty 3605045

CS3103 Rick Wightman

March 14, 2021

[**Project idea**](#_kn0qp8hq7vka) **1**

[**Some user stories to help guide this assignment:**](#_at6f36pce747) **1**

[**ERD**](#_yonr2kmzwtlt) **2**

[**Dictionary of stored procedures (names only) and their functionality:**](#_ykmjzskg82vk) **2**

[Create](#_fh45ippaov49) 3

[Read](#_wsy5grbjsxyz) 3

[Update](#_2gbpl91eyn2q) 3

[Delete](#_ujy8u7ocrwrp) 3

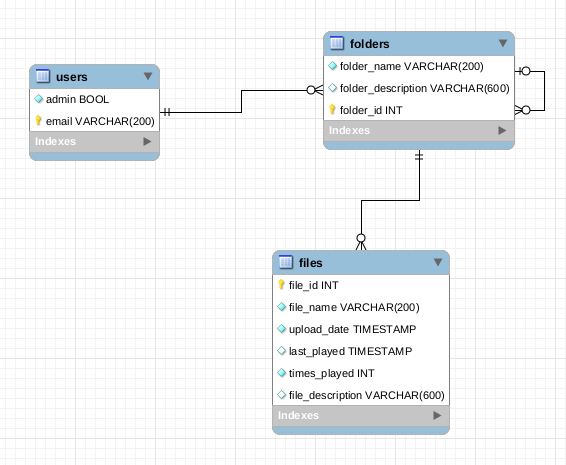
# Project idea

A personal online audio storage/streaming service.

# Some user stories to help guide this assignment:

* Create personal account
* Create admin account
* Log into account
* Change email
* Create folders
* Change folder name
* Organize folders into folders
* Upload audio files
* Download audio files
* View all uploaded audio files
* Change audio file name
* Organize audio files into groups (folders)
* Delete audio files
* Delete account
* Delete other accounts (admin)
* Delete other users’ data (admin)
* Delete other users’ files (admin)
* View number of times file played
* View number of times file was downloaded/saved
* View upload date of file
* View creation date of folder
* Play and pause audio file in browser with volume control

# ERD



Notes:

* The file\_name includes the extension indicating format
* audio files will not be stored in a database, just their metadata
* Each user has a root folder, and the root folder has child folders, and so on.
* The UNB CS system will store username and password used to access many parts of the system

# Dictionary of stored procedures (names only) and their functionality:

Please see the sql for full details

## Create

* addUser: given email, creates a user for the audio platform
* addFolder: given folder ID, folder name, and folder description, creates a folder with that name under that folder.
* addFile: given file name, file description, folder ID, adds a file with that metadata and a foreign key to that folder. times\_played is set to 0 and the current system time is set as upload\_date.

## Read

* findFolderString : given folder substring and user email, return all matching user folder IDs
* getUser: given user email, returns admin status
* getUsers: returns list of non admin users (admin only access)
* getFolder: given folder ID, returns folder name, parent folder (0 if root) and descriptive text (such as folder genres, number of songs)
* getFile : given file ID, returns file name, parent, upload time, number of plays, last played time and descriptive text (such as genre, length, popularity)
* findFileString : given file substring and user email, return all matching user file IDs

## Update

* setFolder : given folder ID and text, replaces description and name
* setFile : given file ID, file description, file name, parent, file last played time, file number of plays, replaces everything except the fileID
* setUser : given old user email, new user email, and admin status, changes user

## Delete

* deleteFolder : given folder ID, deletes given folder and all contents, except the root folder
* deleteUser : given user email, deletes user records and all user folders and files, including root folder
* deleteFile : given file ID, the audio file will be removed