Scientific Python Interactive Data Acoustic Modeling (SPIDAM) Project

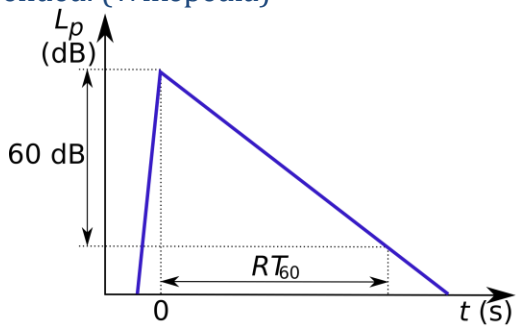
Background

Student groups will create a comprehensive interactive data analysis and modeling platform that enables users to seamlessly import, clean, visualize, analyze, and model data using scientific calculations and data visualization tools.

The platform will tackle the problem of voice intelligibility in enclosed spaces that can be disruptive to hearing and learning. Large enclosed spaces, if not properly treated, may induce long reverberation times.

Reverberation (commonly shortened to **reverb**), in acoustics, is a persistence of sound after it is produced.[1] Reverberation is created when a sound or signal is reflected. This causes numerous reflections to build up and then decay as the sound is absorbed by the surfaces of objects in the space – which could include furniture, people, and air.[2] This is most noticeable when the sound source stops but the reflections continue, their amplitude decreasing, until zero is reached. (Wikepedia)

Reverberation time is a measure of the time required for the sound to "fade away" in an enclosed area after the source of the sound has stopped.
When it comes to accurately measuring reverberation time with a meter, the term T_{60} [3] (an abbreviation for reverberation time 60 dB) is used. T_{60} provides an objective reverberation time measurement. It is defined as the time it takes for the sound pressure level to reduce by 60 dB, measured after the generated test signal is abruptly ended. (Wikepedia)



The longer the reverb decay, the more voice intelligibillity will be impaired making comprehension and understanding difficult for the listener. Due to the sound absorption coefficient of surface materials (floor, walls, ceiling), different frequencies (low tones vs higher tones) may have different reberb times. Identifying the frequency ranges that have the longest reverb times is necessary to engineer a solution. The solution will enable the implementation of the appropriate acoustic treatment for dampening specific frequency ranges with the longest reverb times. The goal is to create a short (< .5 second), yet more consistent reverb time over the audible frequency spectrum (20hz – 20khz).

Students will select and enclosed space on campus that has at least RT60 > 1 second (of reverb time). Examples could be an IST hallway or Aula Magna.

Students will record the sound of a human clap from a distance of 3 meters using a mobile phone, laptop or dedicated audio recording device to analyze using Python and libraries such as Pandas, SciPy, Matplotlib, Numpy, or LibROSA.

Student application must utilize a graphical user interface such as tkinter to:

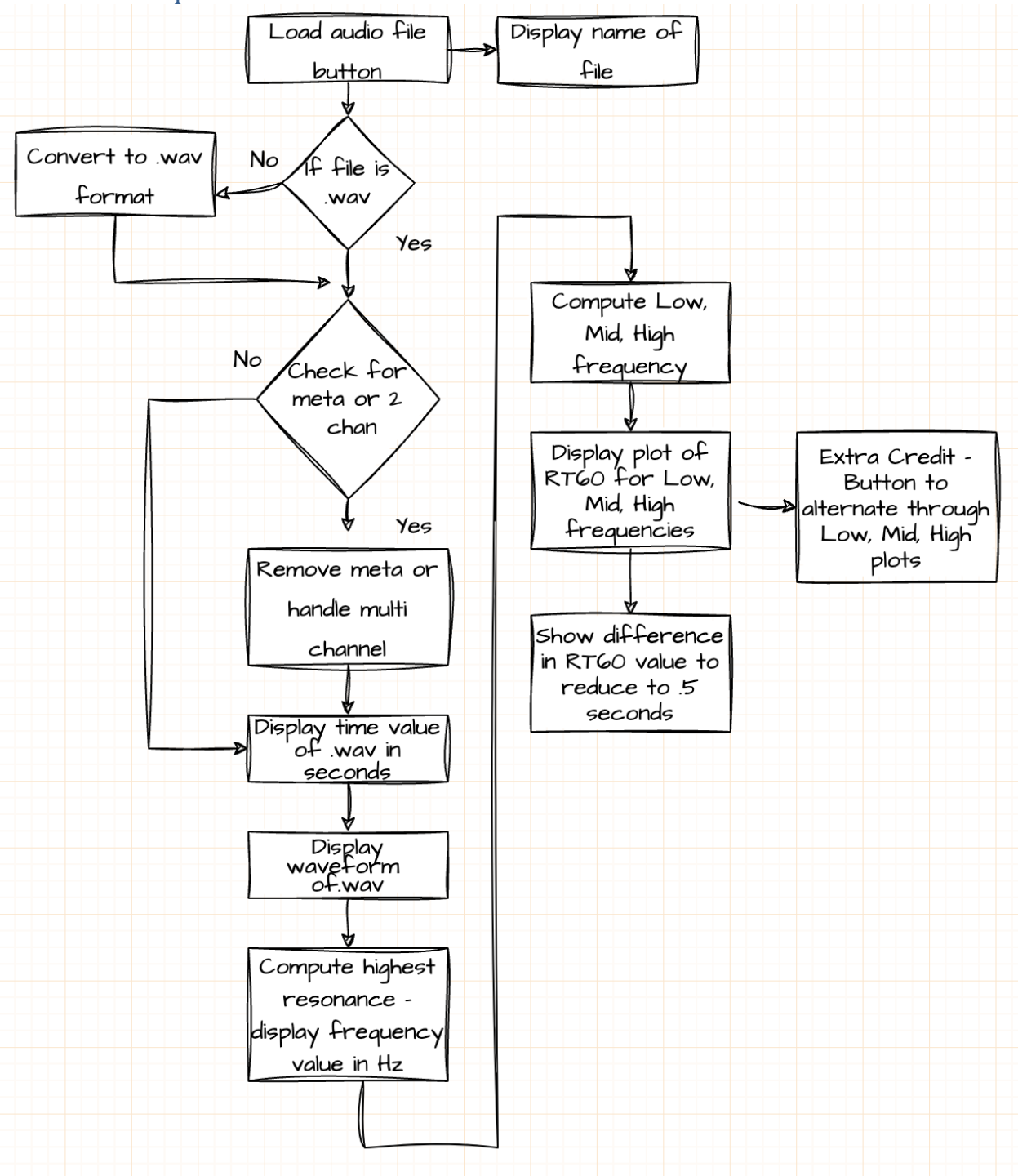
- Import data
  - Load data file from various sources
- Clean
  - Provide data cleaning tools to handle missing values, remove meta data, and inconsistencies in channel format.
  - Detect and correct data formatting issues, such mp3 to wav format conversion
- Data Analysis
  - Generate summary statistics and descriptive measures such as length of audio sample and RT60 value
  - Create data visualizations, including histograms, scatter plots, boxplots, or bar charts. Visualize the original waveform
  - Identify patterns or trends in the data, specifically RT60 values over three frequency ranges
- Data modeling
  - Visualize model performance and interpret results
  - Visualize RT60 data for each frequency range
  - Display the greatest resonant frequency (greatest value)
- Reporting
  - Identify the difference in RT60 time to reduce RT60 to maximum voice intelligibility of .5 seconds

Student group will also present a report and oral presentation of their project and discoveries.

Procedure

Create a Python program with a graphical user interfaces that measures and analyzes reverberation time (RT60) utilizing the api for Pandas, Scipy or equivalent to aid in scientific computations.

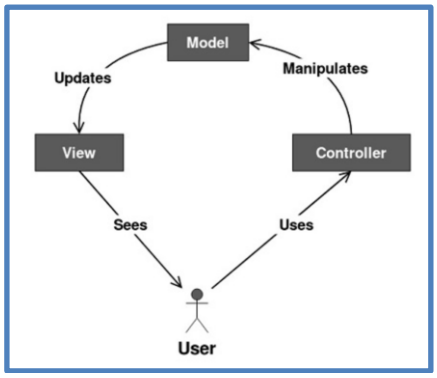
Functional Requirements:



Design Requirements:

Use a Model – View – Controller design pattern for architecture of Python files.

- **State information** is kept in **model** classes. These are the items being viewed and manipulated.
- The **view** is how the model is **presented and interacted** with by the user and are things that will change.
- The **controller** is where the **flow of the application** is managed. All the **sequencing of interactions** between the user and the system is controlled by these modules. See Tutorial Point



Version Control:

Utilize Git/Github to manage source control and check out branches by group members. Provide public url to your Github repository. Gitlog to be provided. Gitlog should indicate weekly activity and equal distribution of commits by group members. Always check a branch then merge to master if it is acceptable.

Report Requirements:

See Final Report template attached.

Python program includes:

- README.md - describing the project, why it exists, and basic usage instructions. Plain text or markdown.
- CHANGELOG.md - summarizes the changes made in each release of your project.
- requirements.txt – list of packages that need to be pip installed
- setup.cfg - configuration file for setuptools. It is an INI style configuration file with key = value entries organized into sections, each of which starts with a name in square brackets, for example [metadata].
- .py modules named and organized according to MVC pattern.

Summary of Group Deliverables:

- Python application in proper format - zipped parent directory
- Audio sample data in wav format (in above zip)
- Gitlog
- Written report

Rubric Category	Points
Functionality – Does the code work?	15
Organization - Is the code clean and organized?	5
Change - Effective use of Version Control	10
Evaluate – Employ definitive condition checks for cleaning validating input	5
Information Literacy– Proper documentation of code, Classes and README.	5
Creativity – Graphical User Interface is creative and effective design	5
Teamwork – Balanced and equitable share of team work.	25
Completeness – All requirements of the assignment are complete	10
Conciseness – Group report meets all requirements and is written professionally	20
Extra Credit: button implementation that toggles gui display of low, mid, high RT60 plots	5

References:

1. ***Valente, Michael; Holly Hosford-Dunn; Ross J. Roeser (2008). Audiology. Thieme. pp. 425–426. ISBN 978-1-58890-520-8.***
2. ***Lloyd, Llewelyn Southworth (1970). Music and Sound. Ayer Publishing. pp. 169. ISBN 978-0-8369-5188-2.***
3. ***"Reverberation Time RT60 Measurement". www.nti-audio.com.***
4. ***Bistafa SR, Bradley JS. Reverberation time and maximum background-noise level for classrooms from a comparative study of speech intelligibility metrics. J Acoust Soc Am. 2000 Feb;107(2):861-75. doi: 10.1121/1.428268. PMID: 10687696.***
5. ***https://en.wikipedia.org/wiki/Reverberation***