Decibel Threshold Event Displayer

BTI3031 Project 1 | Final Presentation

January 8, 2025

Dominic Gernert, Lukas von Allmen, Darius Degel

Table of Contents

Initial Situation

Project Goals

- Analyze Audio File

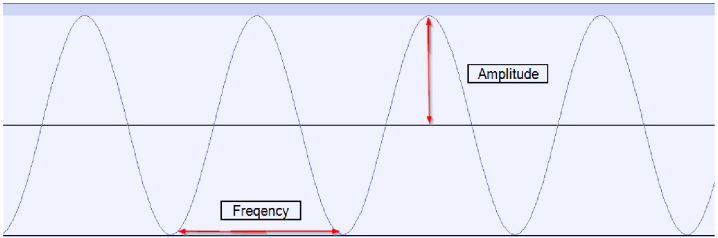
Project Goals

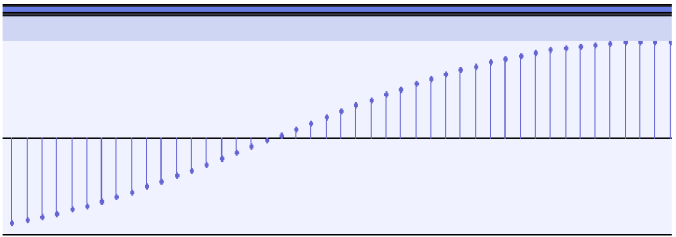
- Analyze Audio File
- Summarize findings in a PDF

Project Goals

- Analyze Audio File
- Summarize findings in a PDF
- Easy to use

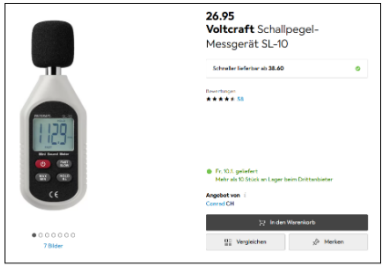
Audio Files





Measuring the Sound Level

DecibelX for IOS



Sound level measuring device from Galaxus

Requirements

- Take .wav file, threshold and additional reference values as input

Requirements

- Take .wav file, threshold and additional reference values as input
- Analyze and Summarize

Requirements

- Take .wav file, threshold and additional reference values as input
- Analyze and Summarize
  - Metadata

Requirements

- Take .wav file, threshold and additional reference values as input
- Analyze and Summarize
  - Metadata
  - Plot

Requirements

- Take .wav file, threshold and additional reference values as input
- Analyze and Summarize
  - Metadata
  - Plot
- User should not need any Technical know-How

Requirements

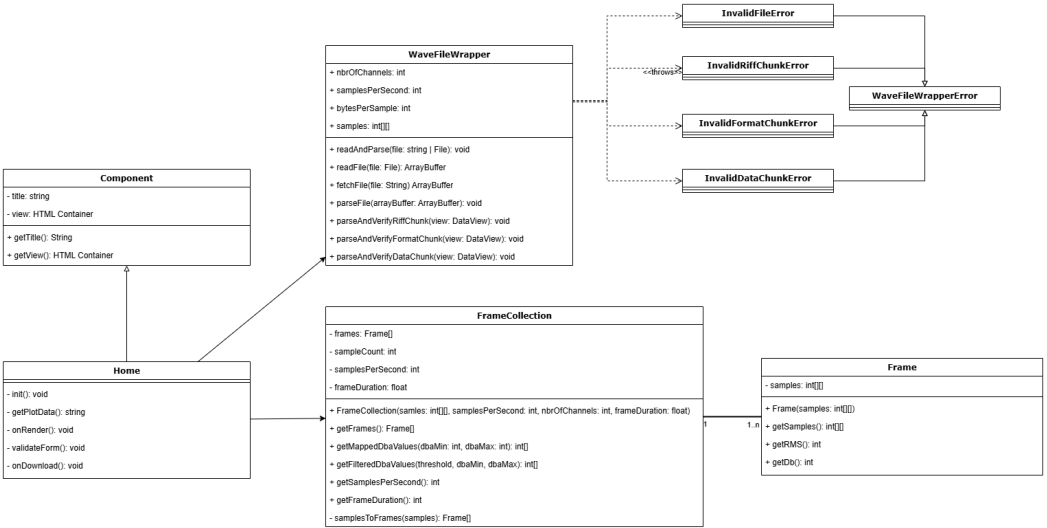
- Take .wav file, threshold and additional reference values as input
- Analyze and Summarize
  - Metadata
  - Plot
- User should not need any Technical know-How
- Platform independent

Technology evaluation

Technology	Total score
Kotlin minimal	74
Kotlin bundled	56
Web SwiftLaTeX	82

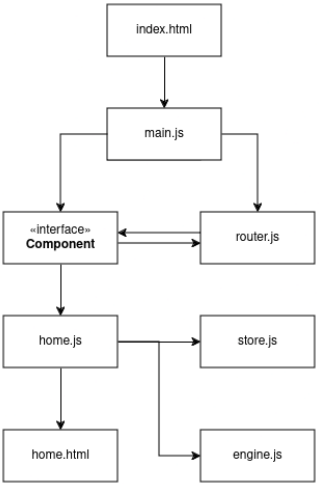
Table of Contents

Architecture - Class Diagram



Architecture - SPA Techstack

- Vanilla JS SPA Framework (Web Programming Module)
- Bootstrap CSS Framework
- SwiftLaTeX in Browser WASM LaTeX rendering Library

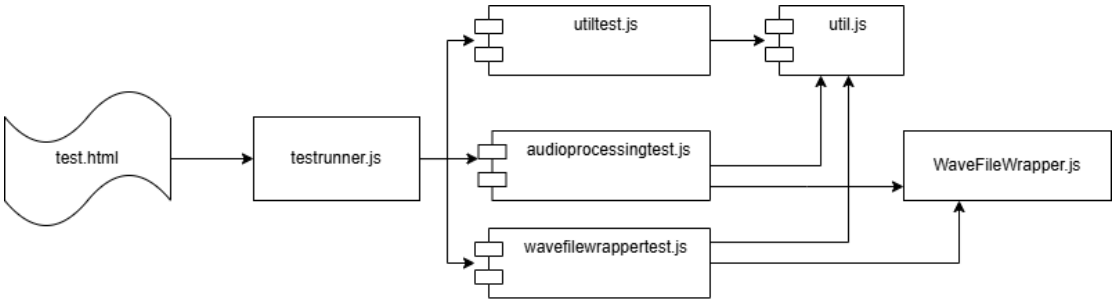


Processes

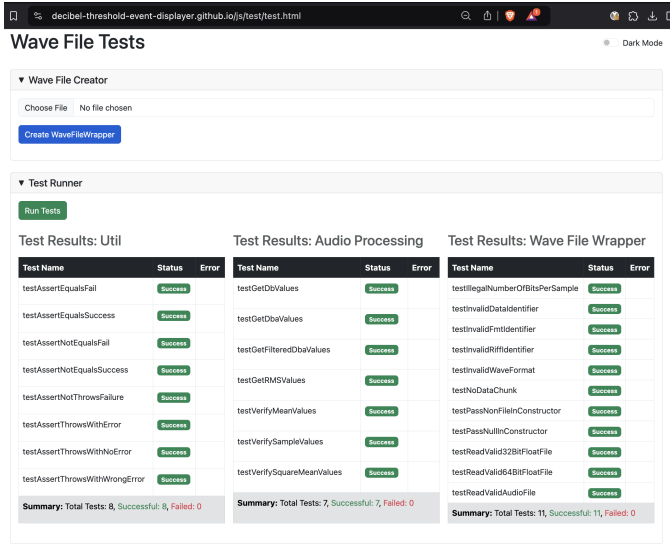
1. Read *.wav File
2. Group samples into frames (duration of 300ms)
3. Calculate root-mean-square (RMS) per frame
4. Convert RMS dB values per frame
5. Map the relative dB to absolute dB(A)
6. Filter the resulting list of dB(A)
7. Render PDF with dB(A) and user data

```
function rms(values){
  const squared = values.map(
      sample => Math.pow(sample, 2)
  );
  const sum = squared.reduce((a, b) => a + b);
  const mean = sum / values.length;
  return Math.sqrt(mean);
}

function rmsToDb(rms) {
  return 20 * Math.log10(rms);
}

function dbToDba(db, dbMin, dbMax, dbaMin, dbaMax) {
  return (db — dbMin) * (dbaMax — dbaMin) /
         (dbMax — dbMin) + dbaMin;
}
```

Testing - Overview

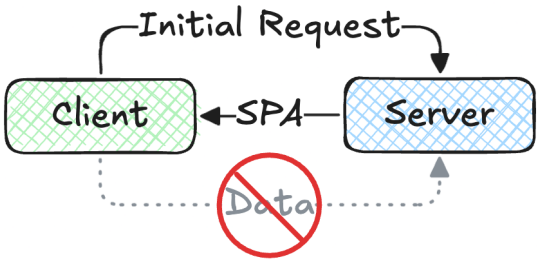


Testing - In action



Privacy concerns

- No data is sent to the server, after the initial request
- From the Plot on the PDF the original Audio File cannot be recreated



The user does not get into legal trouble, using the application or the resulting PDF!

License

Dependency Licenses:

- SwiftLaTeX: AGPL-3.0
- pgfplots: GPL-3.0

Resulting License:

- **GPL-3.0 licence (FLOSS)**

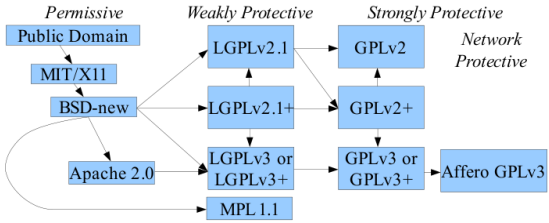


Image Source: https://dwheeler.com/essays/floss-license-slide.html

Deployment / Distribution

1. A dev pushes or merges code to the main branch
2. GitLab automatically mirrors the repository to GitHub
3. GitHub deploys automatically to GitHub Pages
4. The Application is available under: https://decibel-threshold-event-displayer.github.io/

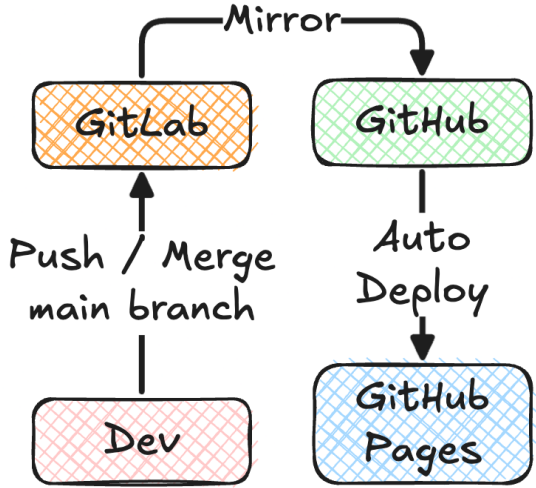


Table of Contents

Scrum

- 2 week iterations
- Daily every week
- Review every other week
- Product goals / sprint goals
- GitLab, MS Teams, LaTeX, excalidraw, draw.io

Table of Contents

Demo

- Watch the demo on YouTube
- Or better yet: Try it yourself on Github Pages!

Table of Contents

Conclusion

- Product goals achieved
- Scrum generally applied
- Minor issues with GitLab
- Great teamwork and team chemistry
- Interesting and well-defined project
- Lots of insights about audio, WAV files, and WebAssembly

Future Work

- Localization (DE, FR, IT)
- Custom thresholds
- Custom form fields
- Support more audio formats
- Dark mode / visual improvements

Questions / Discussion