Song Transposer Project Report

This report outlines the development of the Song Transposer project, a software system designed to analyse and transpose audio recordings (ideally of songs) into different musical keys. The project spans multiple domains, including digital signal processing (DSP), front-end development, back-end services, and full-stack architectural design. It also covers the selection of technologies, file organisation, applied algorithms, and development practices followed throughout the project lifecycle. Limitations of the system and possible areas for future enhancement are also discussed.

1. Introduction

The aim of the Song Transposer project was to implement an end-to-end tool capable of accepting an audio file of a song, estimating its musical key, and transposing the audio to a user-specified target key. The application integrates a Python back end for signal analysis and processing, a JavaScript front end for user interaction, and well-defined communication interfaces.

2. System Architecture

The system follows a modular design comprising three components: a Vue.js front end, a FastAPI back end, and a DSP module for audio analysis and transformation.

3. Technology Stack

The project employs: FastAPI for the back end, Vue.js for the front end, Librosa and SciPy for DSP, the Krumhansl–Schmuckler algorithm for key detection, and FFmpeg for audio format handling. These technologies were selected for performance, ecosystem maturity, and suitability for audio analysis.

4. Algorithms and DSP Concepts

The Krumhansl–Schmuckler algorithm is used for key detection by comparing chroma vectors to empirical key profiles. Pitch-shifting is implemented using a phase-vocoder approach, manipulating frequency-domain representations via the short-time Fourier transform (STFT). Spectral peak detection supports refined pitch-class estimation.

5. Limitations

Key detection may be inaccurate for highly chromatic or modulatory music. Real-time processing is not currently supported. Artefacts may occur for extreme pitch-shifts. Visual feedback such as spectrograms is not implemented.

6. Future Work

Future improvements include real-time transposition, batch processing support, improved UI visualisation, neural key-detection models, automated testing, and cloud scalability.

7. Conclusion

The Audio Transposer project integrates DSP methods, full-stack development, and computational musicology. It demonstrates the application of architectural design, algorithmic audio processing, and established engineering practices in building a functional musical tool.