

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЧНОЇ ТРАНСКРИПЦІЇ ТА НОТУВАННЯ ФОРТЕПІАННОЇ МУЗИКИ

Вінницький національний технічний університет

Анотація

У роботі розглянуто процес архітектурного проєктування веб-орієнтованої інформаційної системи автоматичної транскрипції та нотування фортепіанної музики. Систему реалізовано у вигляді кросплатформеного вебзастосунок на мові Python із використанням фреймворку Streamlit та побудовано за модульним принципом. Розроблено гнучку архітектуру, що охоплює модулі завантаження й попередньої обробки аудіосигналу, спектрального аналізу, розпізнавання нотних подій, генерації MIDI-файлу та формування підсумкового нотного запису у форматах HTML і PDF. Для частотного аналізу сигналів використано FFT/STFT-перетворення, алгоритм YIN для визначення висоти тону в монофонічному режимі та метод гармонічного сканування спектра для поліфонічної транскрипції. Запропонована інформаційна система може слугувати ефективним інструментом підтримки освітнього процесу для студентів музичних спеціальностей, викладачів та дослідників.

Ключові слова: інформаційна система, вебзастосунок, автоматична транскрипція музики, нотування партитур, Streamlit, короткочасне перетворення Фур'є, алгоритм YIN, MIDI, модульна архітектура.

Abstract

The paper considers the architectural design process of a web-oriented information system for automatic transcription and notation of piano music. The system is implemented as a cross-platform Python-based web application using the Streamlit framework and follows a modular architecture. The developed architecture includes modules for audio loading and preprocessing, spectral analysis, note event recognition, MIDI file generation, and score rendering in HTML and PDF formats. FFT/STFT transformations, the YIN pitch detection algorithm for monophonic mode, and harmonic spectrum scanning for polyphonic mode are used for digital audio analysis. The proposed information system serves as an effective educational support tool for music students, teachers, and researchers.

Keywords: information system, web application, automatic music transcription, score notation, Streamlit, Short-Time Fourier Transform, YIN algorithm, MIDI, modular architecture.

Вступ

Автоматична транскрипція музики є актуальним напрямом розвитку інформаційних систем і технологій, оскільки дає змогу перетворювати неструктуровані аудіозаписи у структуроване цифровізоване нотне подання. Особливо складним і важливим це завдання є для поліфонічної фортепіанної музики, де якість результату залежить від точності аналізу частотного спектра, визначення висоти тону, тривалості нот і динамічних моментів їх появи.

Ручне нотування музики «на слух» потребує значного часу, фахового досвіду та часових витрат. Для студентів музичних спеціальностей, педагогів і початківців автоматизований веб-інструмент транскрипції може стати ефективним засобом навчання, самоперевірки та оперативного формування нотного матеріалу. Наявні комерційні аналоги часто є платними або функціонують за принципом «чорної скриньки», не надаючи користувачу доступу до проміжних етапів аналізу даних. Це обмежує їх використання в навчальних цілях, де критично важливо демонструвати, як саме аудіосигнал перетворюється на нотні події.

Метою роботи є проєктування кросплатформеної інформаційної системи автоматичної транскрипції та нотування фортепіанної музики, яка забезпечує повний конвеєр обробки: від завантаження аудіофайлу до генерації стандартизованих MIDI-файлів та інтерактивного рендерингу нотного запису у зручному для користувача вигляді.

Результати дослідження

У процесі проєктування інформаційної системи було визначено її основні функціональні компоненти та інформаційні потоки між ними. Система реалізована як вебзастосунок на мові Python із використанням фреймворку Streamlit, що забезпечує простий інтерфейс для завантаження аудіофайлів, вибору параметрів аналізу та перегляду результатів транскрипції.

Архітектура системи побудована за модульним принципом. Центральним компонентом є головний модуль `app.py`, який ініціалізує інтерфейс, керує вкладками, зчитує параметри користувача, послідовно викликає інші модулі та зберігає проміжні результати у стані сесії Streamlit (рис. 1). Такий підхід дає змогу розділити відповідальність між компонентами та спростити подальше тестування й модифікацію системи.

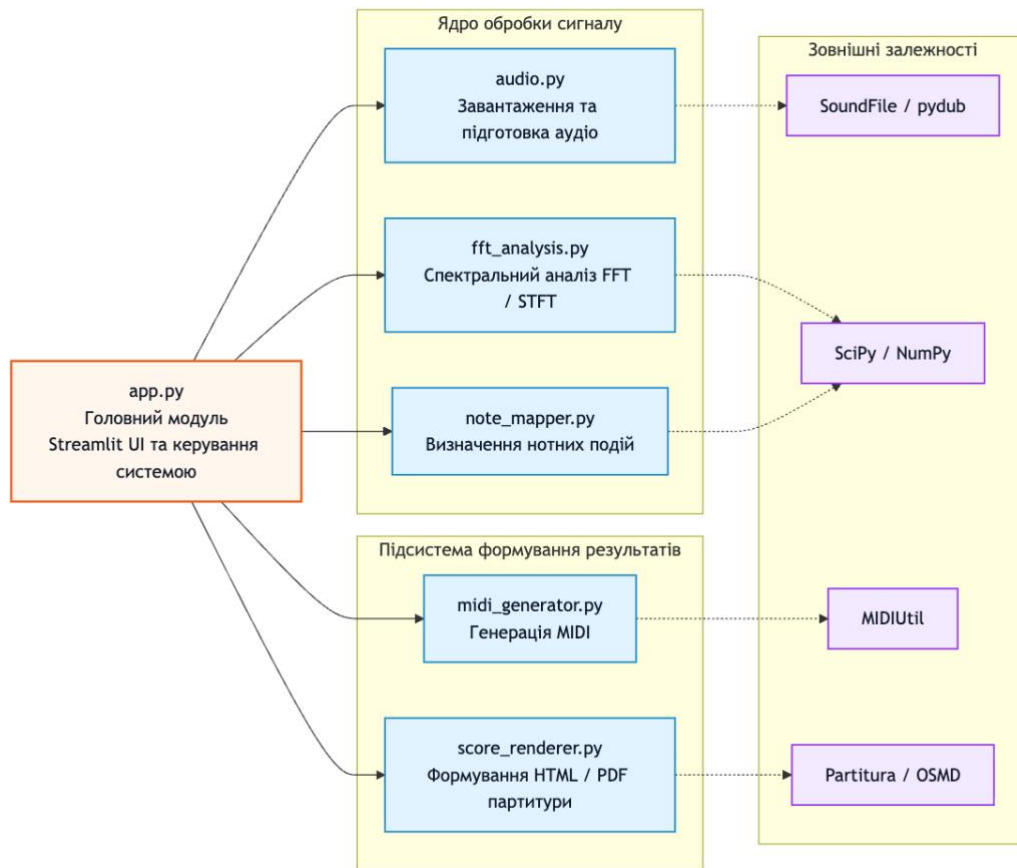


Рис.1 – Модульна архітектура інформаційної системи автоматичної транскрипції музики

Першим етапом обробки є завантаження та підготовка аудіосигналу, що реалізовано в модулі `audio.py`. Модуль приймає вхідний аудіобуфер у підтримуваних форматах, за потреби виконує конвертацію у WAV, зчитує дані у вигляді масиву `float32`, перетворює стереосигнал у моно шляхом усереднення каналів, усуває постійну складову, виконує фільтрацію та нормалізацію сигналу. Це дозволяє підготувати аудіодані до подальшого спектрального аналізу.

Наступним етапом є спектральний аналіз, який виконується модулем `fft_analysis.py`. У системі застосовано два основні підходи: одноразове FFT-перетворення для побудови загального спектра фрагмента та STFT-перетворення для отримання спектрограми. FFT дає змогу оцінити загальну частотну структуру аудіофайлу, тоді як STFT формує тривимірне подання амплітуд у координатах «частота — час», що використовується для розпізнавання нотних подій.

Ключовим компонентом системи є модуль `note_mapper.py`, який відповідає за інтерпретацію спектральних даних у нотні події. У системі передбачено два режими роботи: `Melody` та `Chords`. У режимі `Melody` використовується алгоритм YIN, який визначає основну частоту сигналу на основі кумулятивної нормалізованої різницевої функції. Після цього виконуються постобробні кроки: медіанна фільтрація, корекція октавних стрибків, округлення до найближчого MIDI-номера, голосування більшістю та об'єднання суміжних сегментів.

У режимі `Chords` для кожного кадру STFT виконується гармонічна оцінка MIDI-кандидатів у діапазоні від A0 до C8. Перед оцінюванням застосовується відбілювання спектра, що зменшує вплив тембру та

дозволяє нотам у різних регістрах конкурувати на рівніших умовах. Для кожного кандидата враховується наявність перших гармонік, нормована амплітуда основного тону, гармонічна підтримка та кількість підтверджувальних гармонік. Після відбору кандидатів застосовуються фільтри гармонічних витоків, октавних дублікатів і надто близьких нот.

Для формування MIDI-файлу використано модуль `mid_i_generator.py`. Він перетворює список нотних подій у стандартне MIDI-подання з урахуванням темпу, часу початку та тривалості нот. У результаті користувач отримує MIDI-файл, який може бути відкритий у секвенсорах, музичних редакторах або інших програмах для подальшого редагування.

Окремий модуль `score_renderer.py` відповідає за формування нотного запису. Він виконує квантизацію нотних подій, оцінює тональність, будує XML-партитуру за допомогою бібліотеки `Partitura` та забезпечує рендеринг результату у форматах HTML і PDF. Для відображення нотного запису використовується `OpenSheetMusicDisplay`, а для створення PDF-звіту — `Playwright/Chromium`.

Для формального опису архітектури системи було розроблено UML-діаграми. Діаграма компонентів демонструє модульну структуру системи та залежності між основними програмними компонентами. Діаграма розгортання показує, що користувач взаємодіє із системою через веббраузер, тоді як обробка аудіоданих виконується в середовищі Python на сервері застосунку (рис. 2).

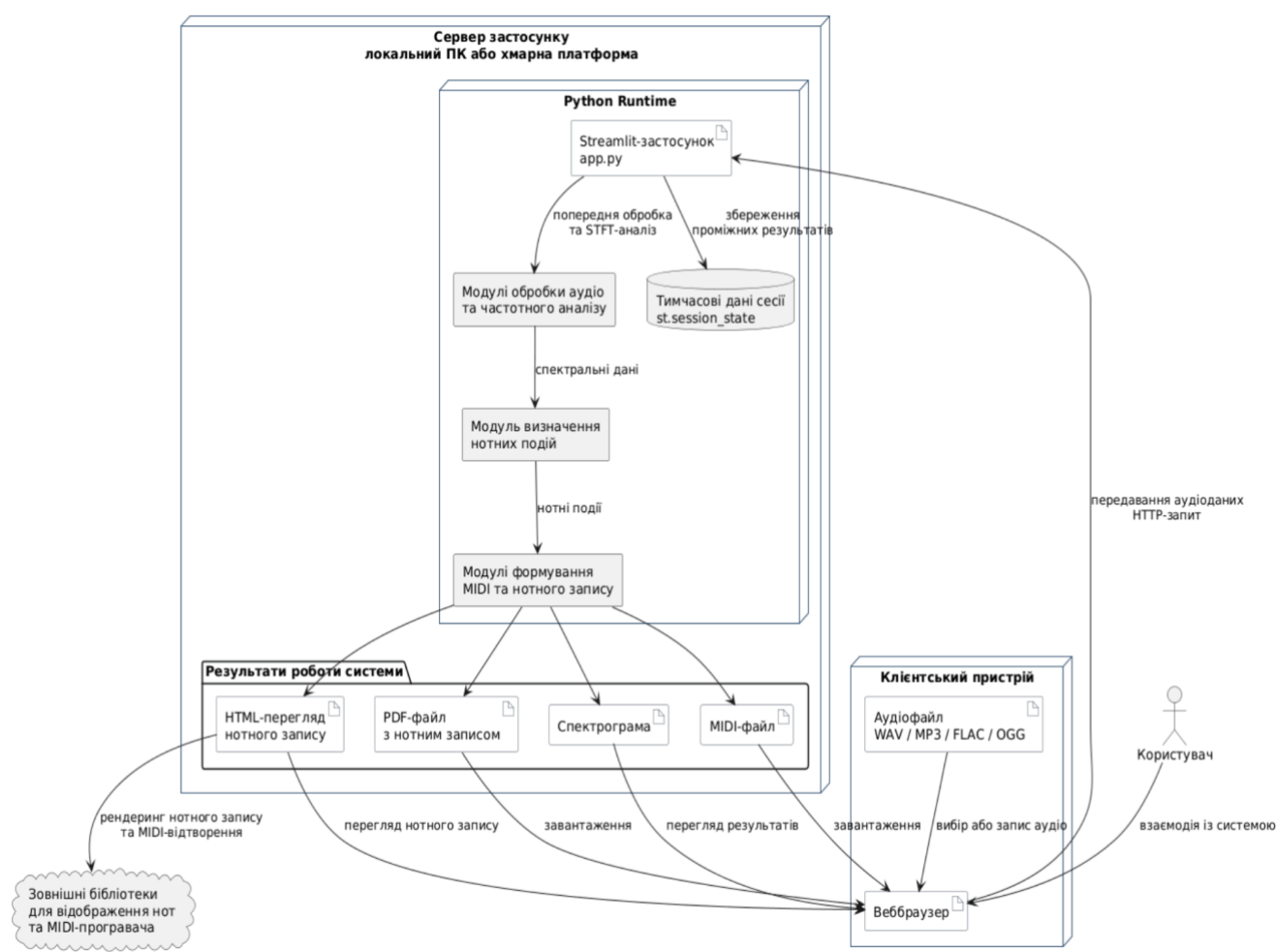


Рис.2 – Діаграма розгортання інформаційної системи

Діаграма послідовності відображає основний сценарій роботи: завантаження аудіофайлу, передавання даних у модуль обробки, обчислення STFT/FFT, розпізнавання нот, генерація MIDI та формування HTML/PDF-партитури (рис. 3).

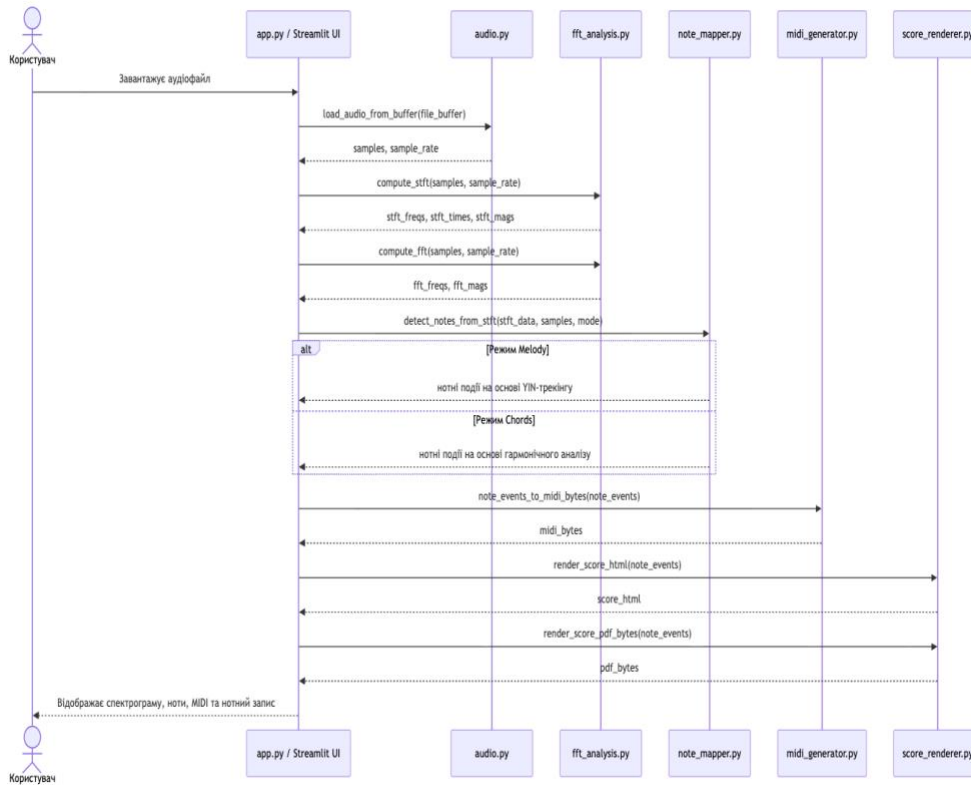


Рис.3 – Діаграма послідовності основного сценарію роботи системи

Діаграма класів описує ключові структури даних, зокрема словник нотної події NoteEvent, який містить поля частоти, назви ноти, октави, MIDI-номера, часу початку, тривалості та рівня впевненості. Також описано допоміжні структури для побудови нотної сітки та формування об'єктів партитури перед експортом у MusicXML.

Діаграми діяльності деталізують алгоритми розпізнавання нот у режимах Melody та Chords. Для режиму Melody алгоритм охоплює обчислення YIN-CMNDF, порогову фільтрацію, згладжування треку висоти тону, корекцію октавних стрибків, визначення повторних атак і формування нотних сегментів. (рис. 4)



Рис.4 – Діаграма діяльності для режиму Melody

Для режиму Chords алгоритм передбачає покадрове гармонічне сканування, стабілізацію нотних наборів у часі, побудову часових сегментів та фінальне усунення гармонічних дублікатів (рис. 5).

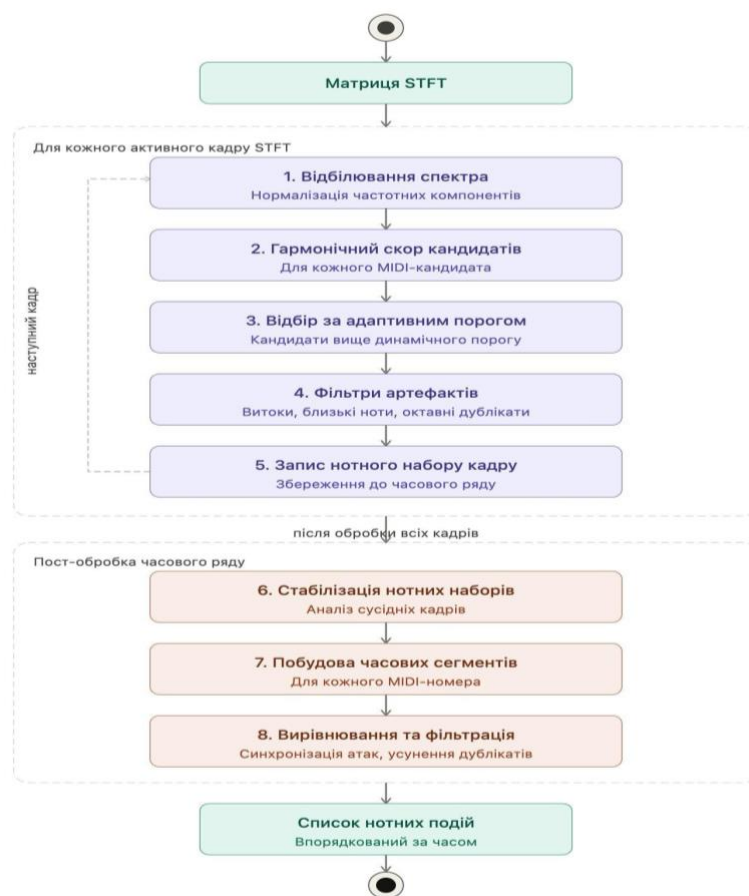


Рис.5 – Діаграма діяльності для режиму Chords

Взаємодія між модулями здійснюється через чітко визначені параметри функцій і повернені значення без прихованого спільного стану. Це підвищує зрозумілість архітектури, полегшує тестування окремих підсистем і створює основу для подальшого розширення системи, зокрема додавання методів машинного навчання, підтримки інших музичних інструментів або розгортання на хмарній платформі.

Висновки

У роботі успішно виконано проектування розподіленої інформаційної системи автоматичної транскрипції та нотування фортепіанної музики. Розроблено модульну безстатусну (stateless) архітектуру, яка раціонально розподіляє відповідальність між головним модулем, блоками попередньої обробки аудіосигналу, спектрального аналізу, розпізнавання нот та генерації вихідних файлів.

Визначено та обгрунтовано ключові алгоритмічні рішення системи: використання FFT/STFT для цифрової обробки сигналів, застосування алгоритму YIN для монофонічного трекінгу та гармонічного сканування з відбілюванням спектра для поліфонічного режиму. Побудовано комплект UML-діаграм (компонентів, розгортання, послідовності та діяльності), які повністю формалізують інформаційні потоки та логіку взаємодії елементів системи.

Практичне значення результатів полягає у створенні архітектурного базису для відкритого програмного інструменту, що дозволяє користувачам безкоштовно отримувати MIDI-дані та PDF-партитури безпосередньо через веб-інтерфейс.

Подальший розвиток системи доцільно спрямувати на інтеграцію з реляційними базами даних для збереження історії сесій, розширення інструментів ритмічної квантизації та впровадження нейромережових моделей транскрипції.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Klapuri A., Davy M. Signal Processing Methods for Music Transcription. New York: Springer, 2006.
2. Müller M. Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. Cham: Springer, 2015.

3. de Cheveigné A., Kawahara H. YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America. 2002. Vol. 111, No. 4. P. 1917–1930.
4. Virtanen T., Plumbley M. D., Ellis D. Computational Analysis of Sound Scenes and Events. Cham: Springer, 2018.
5. Streamlit. Streamlit documentation : вебсайт. URL: <https://docs.streamlit.io/> (дата звернення: 02.05.2026).
6. SciPy. Signal processing — SciPy documentation : вебсайт. URL: <https://docs.scipy.org/doc/scipy/reference/signal.html> (дата звернення: 02.05.2026).
7. NumPy. NumPy documentation : вебсайт. URL: <https://numpy.org/doc/> (дата звернення: 04.05.2026).
8. Partitura. Partitura: Python package for symbolic music processing : вебсайт. URL: <https://partitura.readthedocs.io/> (дата звернення: 04.05.2026).
9. OpenSheetMusicDisplay. OpenSheetMusicDisplay documentation : вебсайт. URL: <https://opensheetmusicdisplay.github.io/> (дата звернення: 05.05.2026).
10. Mido. Mido — MIDI Objects for Python : вебсайт. URL: <https://mido.readthedocs.io/> (дата звернення: 06.05.2026).

Хорошева Олександра Володимирівна — студентка групи 2ICT-22б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: alexa.khorosh@gmail.com.

Жуков Сергій Олександрович — к.т.н., доцент кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, e-mail: sazhukov@gmail.com.

Khorosheva Aleksandra V. — student of group 2IST-22b, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: alexa.khorosh@gmail.com.

Zhukov Serhii O. — Ph.D., Assistant Professor of the Department of Systems Analysis and Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: sazhukov@gmail.com.