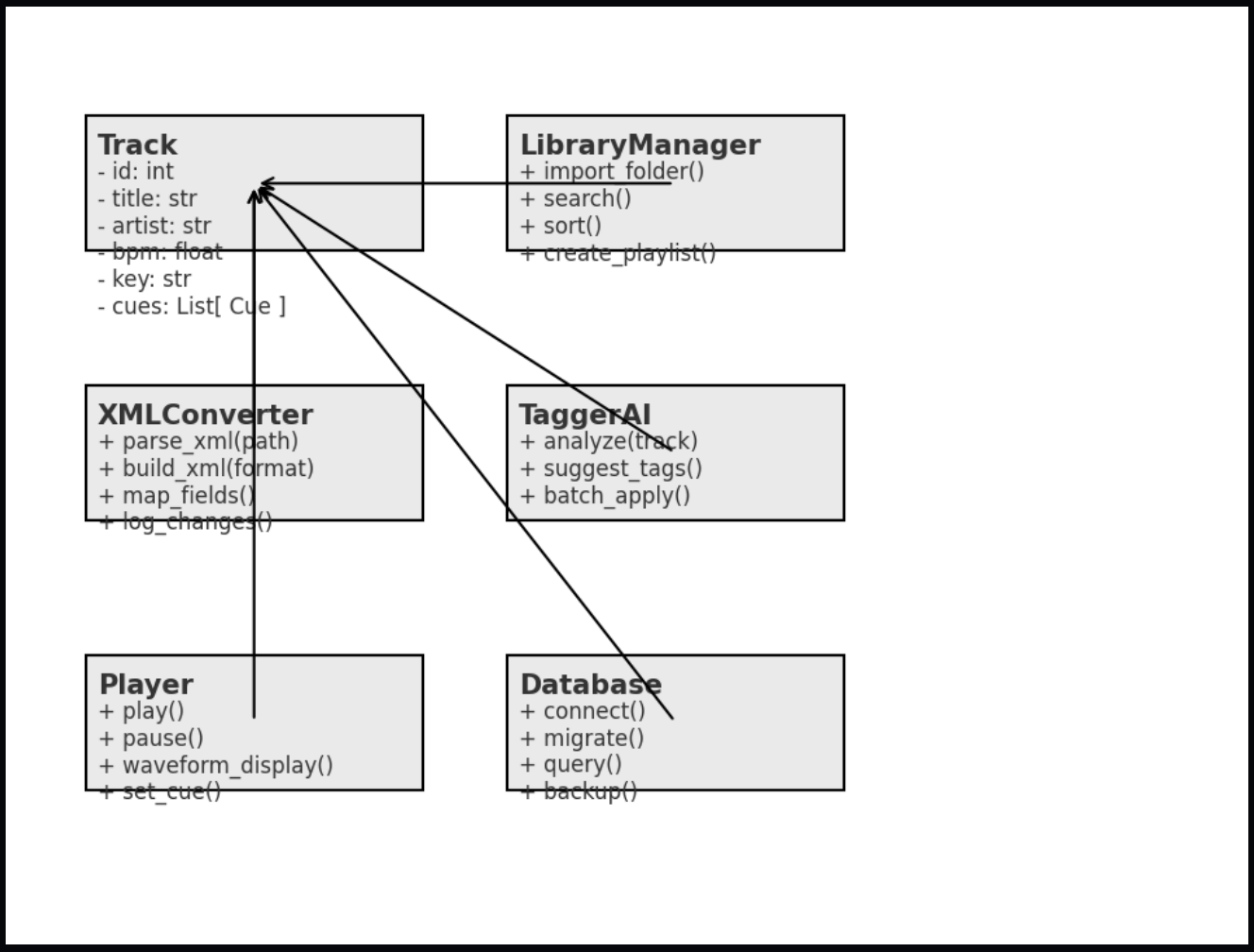
Manager Audio AI — Rozszerzona dokumentacja (UML & ERD, G

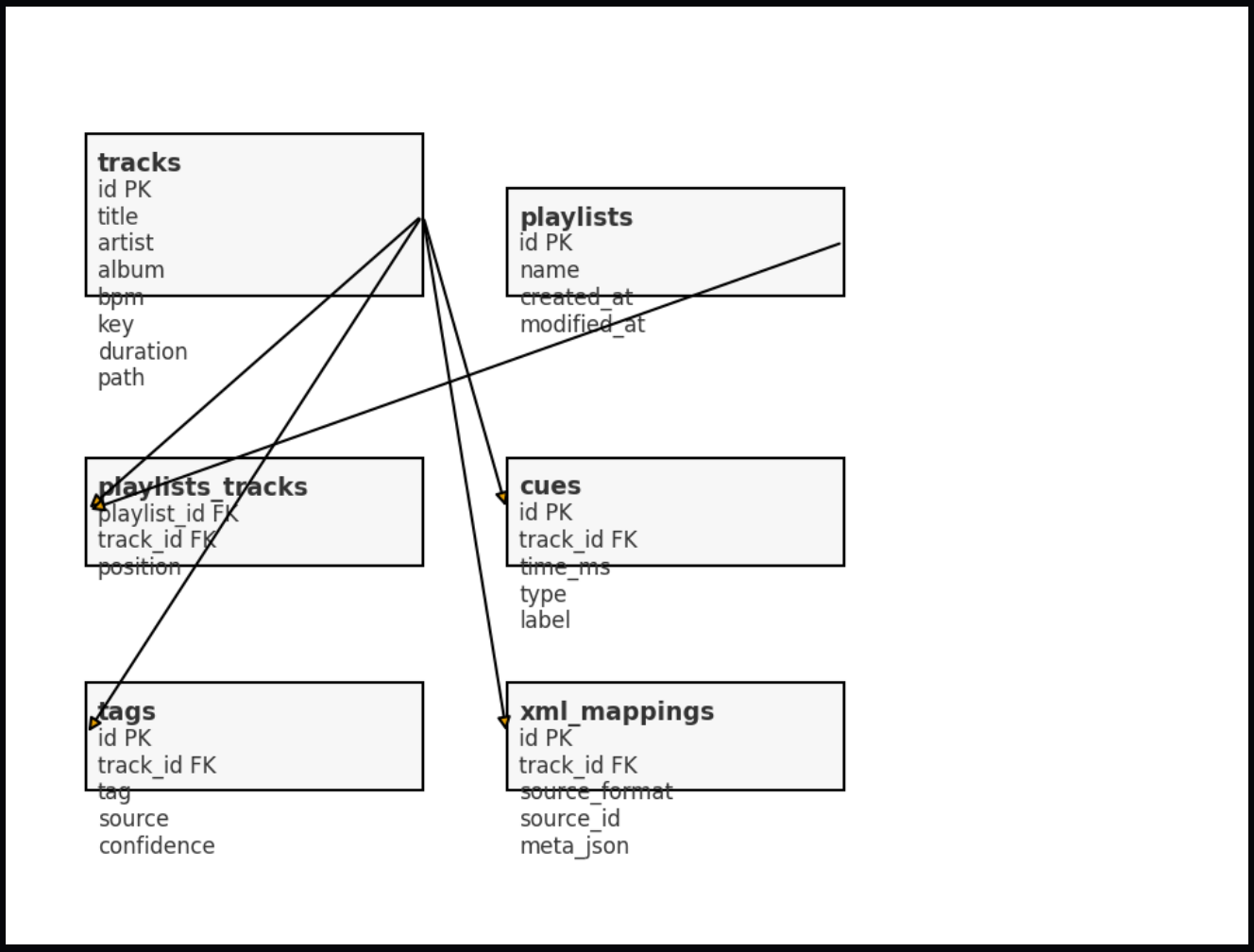
Dodano: diagramy UML, ERD, implementacj■ przyk■adowego GUI (PyQt6) oraz przyk■adowe testy jednostkowe.

Diagram UML — Klasy i relacje



- Diagram klas pokazuje modu■y: Track, LibraryManager, XMLConverter, TaggerAI, Player, Database.
- Ka■da klasa posiada metody publiczne (operacje) oraz atrybuty (dane).
- Strza■ki oznaczaj■ zale■no■ci — wszystkie modu■y pobieraj■/aktualizuj■ dane Track.

ERD — Schemat bazy danych (SQLite)



- G■ówne tabele: tracks, playlists, playlists_tracks, cues, tags, xml_mappings.
- Zalecane indeksy: tracks(path), tracks(bpm), playlists(name), xml_mappings(source_format).
- Migration tool powinien umo■liwia■ dodanie kolumn i wersjonowanie schematu.

Przyk■adowe GUI — PyQt6 (konwerter XML)

```
# PyQt6 example: ConverterWindow (simplified)
from PyQt6.QtWidgets import (QApplication, QWidget, QVBoxLayout, QHBoxLayout,
                             QPushButton, QLabel, QFileDialog, QComboBox, QTextEdit)
from PyQt6.QtCore import Qt
from lxml import etree

class ConverterWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Manager Audio — XML Converter")
        self.resize(900, 500)
        layout = QVBoxLayout()
        # Source selection
        h1 = QHBoxLayout()
        self.src_label = QLabel("Plik ■ród■owy: (brak)")
        btn_src = QPushButton("Wybierz plik")
        btn_src.clicked.connect(self.select_source)
        h1.addWidget(self.src_label)
        h1.addWidget(btn_src)
        layout.addLayout(h1)
        # Format selectors
        h2 = QHBoxLayout()
        self.src_format = QComboBox(); self.src_format.addItems(["Rekordbox","VirtualDJ"])
        self.dst_format = QComboBox(); self.dst_format.addItems(["VirtualDJ","Rekordbox"])
        h2.addWidget(QLabel("Z:")); h2.addWidget(self.src_format)
        h2.addWidget(QLabel("Na:")); h2.addWidget(self.dst_format)
        layout.addLayout(h2)
        # Log area
        self.log = QTextEdit(); self.log.setReadOnly(True)
        layout.addWidget(self.log)
        # Convert button
        btn_convert = QPushButton("Konwertuj")
        btn_convert.clicked.connect(self.convert)
        layout.addWidget(btn_convert, alignment=Qt.AlignmentFlag.AlignRight)
        self.setLayout(layout)
        self.src_path = None

    def select_source(self):
        path, _ = QFileDialog.getOpenFileName(self, "Wybierz plik XML", filter="XML files (*.xml)")
        if path:
            self.src_path = path
            self.src_label.setText(path)

    def convert(self):
        if not self.src_path:
            self.log.append("Brak pliku ■ród■owego.")
            return
```

Przyk■adowe GUI — PyQt6 (cd.)

```
        src = self.src_format.currentText()
        dst = self.dst_format.currentText()
        self.log.append(f"Konwersja: {src} -> {dst}")
        try:
            # simplistic parse & write flow (real logic maps fields)
            tree = etree.parse(self.src_path)
            # map to internal model...
            # build dst xml...
            out_path = self.src_path.replace(".xml", f"_converted_{dst}.xml")
            tree.write(out_path, encoding='utf-8', xml_declaration=True, pretty_print=True)
            self.log.append(f"Zapisano: {out_path}")
        except Exception as e:
            self.log.append(f"B■■d: {str(e)}")

# Run:
# if __name__ == '__main__':
#     app = QApplication([])
#     w = ConverterWindow(); w.show(); app.exec()
```

Przyk■adowe testy jednostkowe (pytest/unittest)

```
# tests/test_converter.py (pytest style)
import os
from your_module import parse_rekordbox_xml, build_virtualdj_xml, Track
def test_parse_and_build(tmp_path):
    sample = tmp_path/"sample.xml"
    sample.write_text('<?xml version="1.0"?><COLLECTION><TRACK
TrackID="1"><TITLE>Test</TITLE><ARTIST>Artist</ARTIST><TEMPO>128</TEMPO></TRACK></COLLECTION>')
    tracks = parse_rekordbox_xml(str(sample))
    assert len(tracks) == 1
    assert tracks[0].title == "Test"
    out = tmp_path/"out.xml"
    build_virtualdj_xml(tracks, str(out))
    assert out.exists()
    txt = out.read_text()
    assert "<VIRTUALDJ" in txt
# unittest example
import unittest
class ConverterUnitTest(unittest.TestCase):
    def test_roundtrip(self):
        # similar to pytest test but using unittest assertions
        self.assertTrue(True)
if __name__ == '__main__':
    unittest.main()
```

Uwaga i nast■pne kroki

- Dalsze rozszerzenia: pe■ne diagramy sekwencji, szczegó■owe migracje DB, przyk■adowe mocki AI oraz
- integracje z serwisami streamingowymi.
- Mog■ przygotowa■ szkielet repozytorium (setup.py/pyproject, requirements.txt, example data) oraz
- skrypt instalacyjny dla Windows/Android.
- Je■li chcesz, wygeneruj■ równie■ assets graficzne makiet w wysokiej rozdzielczo■ci i pliki SVG dla
- diagramów.