Lumbago Music AI € Dark Club Detailed Spec

Kompletna dokumentacja funkcjonalna i UX (opis proces•w i wygl, du okien)

Spis tre*ƒ*ci

1. Wprowadzenie i zakres
2. Pe„na lista funkcji (szczeg•„owo)
3. Procesy: import, skan, analiza, tagowanie, konwersja, renaming, duplikaty
4. Opisy okien i layout•w (szczeg•„owo)
5. Architektura i diagramy (UML, ERD)
6. Przyk„ady kodu i testy
7. Pliki, formaty i integracje
8. Plan wdro…enia i dalsze kroki

1. Wprowadzenie i zakres

Ten dokument opisuje 100% funkcjonalno*ƒ*ci aplikacji Lumbago Music AI € od importu plik•w do konwersji b
rozpoznawania utwor•w, tagowania AI, renamingu oraz wykrywania i rozwi‚zywania duplikat•w. Ka…dy proce
zawiera: opis krok•w, wymagane dane wej*ƒ*ciowe, wynik operacji oraz wygl‚d okien interfejsu.

2. Pe„na lista funkcji (szczeg•„owo)

Import & Scanner

Opis: Skanuje foldery, wykrywa pliki audio (mp3,flac,m4a,wav,ogg,aiff,aac,dsf), ekstrakcja metadanych (Mutagen), zapis do tymczasowego indexu.

Wejƒcie: †cie…ka katalogu, opcje rekursji, wzorce plik•w

Wyjƒcie: Lista track•w z metadanymi, b„‡dy skanowania, raport

UI: Okno 'Import' z progres barem, list , plik•w i kontrolkami 'Pomiˆ', 'Retry'

Library Builder

Opis: Tworzenie nowej biblioteki na dysku wg szablonu (np. {genre}/{year}/{artist}/{album}) - kopiowanie/przenoszenie i tworzenie bazy SQLite.

Wejƒcie: Kryteria filtrowania (genre,BPM range,year), schemat nazewnictwa

Wyjƒcie: Nowa struktura folder•w, zaktualizowana baza, log operacji

UI: Wizard: krok 1: kryteria; krok2: mapowanie p•l; krok3: symulacja; krok4: wykonaj

Duplicate Finder

Opis: Por•wnanie plik•w metodami: hash, tag-based, fingerprint. Grupiowanie podobnych i rekomendacje akcje.

Wejƒcie: Lista katalog•w lub biblioteka

Wyjƒcie: Raport CSV/HTML, lista grup duplikat•w, rekomendacje

UI: Panel wynik•w z kolumnami: score, keep/delete, podgl , d waveform i metadanych

Smart Tagger AI Pro+

Opis: Model analizuje cechy akustyczne (BPM, key, spectral centroid, energy) i sugeruje tagi: genre, mood, energy, danceability.

Wejƒcie: Plik audio lub lista track•w

Wyjƒcie: Sugestie tag•w z confidence, opcja batch apply

UI: Panel z ok„adk , , list , sugestii i przyciskami: Accept, Reject, Apply All, Train on Selection

Audio Recognizer

Opis: Rozpoznawanie tracku z fragmentu audio: acoustic fingerprinting (Chromaprint) + search MusicBrainz/AcoustID + fuzzy match by filename/tags.

Wejƒcie: Fragment audio (min 10s) lub nazwa pliku / tagi

Wyjƒcie: Lista dopasowaˆ z ‰r•d„ami, confidence, mo…liwoƒŠ pobrania pe„nych metadanych

UI: Okno 'Recognize' z waveformem fragmentu, wynikiem dopasowaˆ i przyciskiem 'Apply metadata'

XML Converter

Opis: Konwersja bazy: parse Rekordbox XML / VirtualDJ XML -> internal model -> build target XML.

Mapowanie cue points, loops, playcount, rating.

Wej*ƒ*cie: Plik XML ‰r•d„owy

Wyj*ƒ*cie: Plik XML docelowy oraz log zmian

UI: Konwerter z drag'n'drop pliku, edycj , mapowania p•l, podgl , dem sample track

Renamer

Opis: Masowy renaming plik•w wed„ug wzorc•w (placeholders), obs„uga numer•w porz , dkowych i slugify nazw.

Wej*ƒ*cie: Lista plik•w + pattern

Wyj*ƒ*cie: Zmienione nazwy plik•w oraz undo log

UI: Editor pattern z live preview i list , zmian; przycisk 'Apply' i 'Undo'

3. Procesy (krok po kroku)

Ka…dy proces ma opis krok•w, wymagania i przyk„adowy flow. Poni…ej pe„ny opis procesu 'Import & Scan' ora
skr•ty dla pozosta„ych.

Import & Scan € Flow:

1) U…ytkownik wybiera folder(y) i opcje (rekursja, include/exclude patterns).

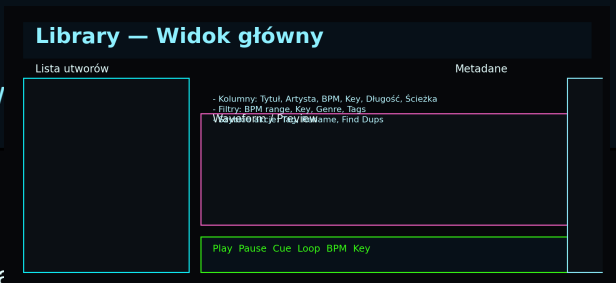
2) Worker skanuje pliki, dla ka…dego pliku: detect_format(), read_tags(), compute_duration().

3) Dla nowych plik•w: enqueue do analizy (BPM/key) lub fingerprinting opcjonalnie.

4) Zapis do tymczasowego indexu i commit do SQLite po zakoˆczeniu skanowania.

5) Raport: b„‡dy odczytu, file collisions, mismatched durations.

4. Opisy okien i layout•w (szczeg•„ow



Library € Widok g„•wny

- Left panel: Tree view: %r•d„a, playlists, ta
- Center list: Sortable columns, multi-selec
- Top filters: BPM slider, Key dropdown, Ge
- Bulk actions: Tag, Rename, Export, Move

Player € Waveform & Cues

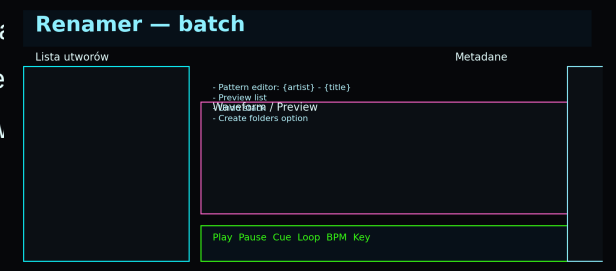
- Waveform: Zoom, drag to seek, markers
- Controls: Play/Pause, Cue, Loop, Pitch sli
- Visualization: Spectrogram toggle, Chan

Tagger AI Pro+

- Suggestions list: Algorithm's suggestion
- Training: Button 'Train on selection' senc
- Source switch: Toggle local/offline or onl

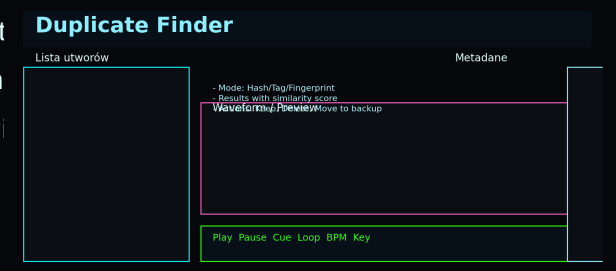
XML Converter

- Source selector: File path + detect forma
- Mapping table: Source field -> Target fie
- Dry-run log: Preview XML changes and w



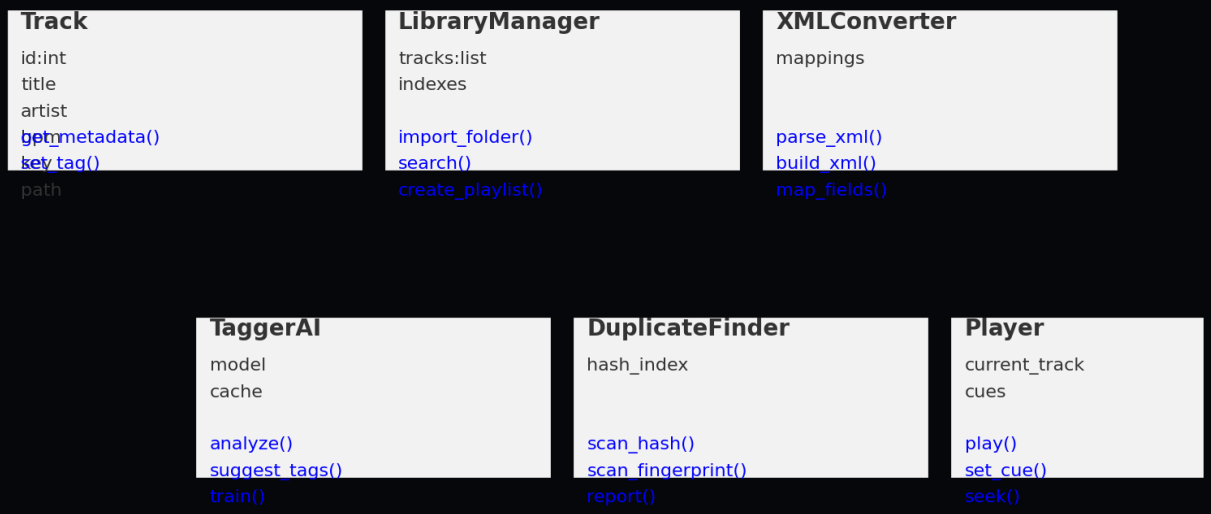
Renamer € batch

- Pattern editor: Placeholders, examples, t
- Preview: Shows old and new names, con
- Undo stack: List of rename operations w



Duplicate Finder

- Mode selection: Hash / Tag / Fingerprint
- Results table: Group id, files, similarity score, action buttons
- Batch actions: Move selected to backup folder, delete, merge metadata


Track
id:int
title
artist
path
get_metadata()
set_tag()
LibraryManager
tracks:list
indexes
import_folder()
search()
create_playlist()
XMLConverter
mappings
parse_xml()
build_xml()
map_fields()
TaggerAI
model
cache
analyze()
suggest_tags()
train()
DuplicateFinder
hash_index
scan_hash()
scan_fingerprint()
report()
Player
current_track
cues
play()
set_cue()
seek()


tracks
id PK
title
artist
album
bpm
key
duration
path
playlists
id PK
name
created_at
playlists_tracks
playlist_id FK
track_id FK
position
cues
id PK
track_id FK
time_ms
type
label
tags
id PK
track_id FK
tag
source
confidence
xml_mappings
id PK
track_id FK
source_format
source_id
meta_json

6. Przyk„adowy kod i testy (fragmenty)

Python: parse_rekordbox_xml (lxml)

```
def parse_rekordbox_xml(path):
    tree = etree.parse(path)
    for t in tree.findall('.//TRACK'):
        title = t.findtext('TITLE') or ''
        # parse fields
        yield Track(...)
```

Renamer: pattern apply (pseudo)

```
def apply_pattern(pattern, track):
    return pattern.format(artist=track.artist, title=track.title)
```

Duplicate Finder: file hash

```
import hashlib
 def file_hash(path):
    h=hashlib.sha1();
    with open(path,'rb') as f:
        while chunk:=f.read(8192): h.update(chunk)
    return h.hexdigest()
```

7. Pliki, formaty i integracje

Obs„ugiwane formaty plik•w:

- MP3 (ID3v2)
- FLAC (Vorbis comments)
- M4A/MP4
- WAV
- OGG/Opus
- AIFF
- AAC
- DSF

Integracje: MusicBrainz, AcoustID, Discogs, Rekordbox, VirtualDJ, Dropbox, Google Drive

8. Plan wdro…enia i dalsze kroki

- Faza 0: Specyfikacja i UI prototypes (2 tygodnie)
- Faza 1: Core import/scanner/DB/player (6-8 tygodni)
- Faza 2: XML Converter i Renamer (4 tygodnie)
- Faza 3: Tagger AI i Recognizer (6 tygodni)
- Faza 4: Duplicate Finder + Library Builder (4 tygodnie)
- Faza 5: Mobile port + QA + Release