



Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Domeniul Calculatoare și Tehnologia Informației
Specializarea Tehnologia Informației

DaisyPlayer

Studenti:
Ghiață Mihaela-Gabriela, Grupa 1410A
Posmangiu Silviu-Andrei, Grupa 1410A

Cuprins

1. Tematica și obiectivele proiectului	3
2. Prezentarea aplicației	3
3. Descrierea tehnică a aplicației	
3.1 Arhitectura aplicației	5
3.2 Backend – PlayerService	5
3.3 Controller	5
3.4 Interfața grafică (UI)	6
4. Modul de utilizare al aplicației	
4.1 Pornirea aplicației	6
4.2 Încărcarea și selectarea melodiiilor	
4.3 Controlul redării audio	6
4.4 Controlul progresului și al volumului	7
4.5 Moduri de redare	7
4.6 Funcționalități suplimentare	7
5. Contributia membrilor echipei	7
6. Dependențele aplicației	7
7. Modul de execuție	8
8. Concluzii și direcții viitoare	8

1. Tematica și obiectivele proiectului

Acest raport prezintă aplicația finală DaisyPlayer, un media player dezvoltat în Python folosind PySide6 (Qt for Python). Scopul principal al proiectului a fost realizarea unei aplicații complete și funcționale, care să ofere utilizatorilor o experiență intuitivă de ascultare și gestionare a fișierelor audio, prin intermediul unei interfețe grafice moderne și a unor funcționalități de redare bine structurate.

Dezvoltarea unei aplicații proprii de tip music player a pornit din necesitatea de a avea un control mai flexibil asupra modului de organizare și redare a melodiilor, fără constrângerile impuse de aplicațiile comerciale existente. DaisyPlayer își propune să ofere un nivel ridicat de personalizare, suport pentru teme vizuale, afișarea metadatelor și posibilitatea extinderii funcționalităților în viitor.

Obiectivele proiectului includ:

- implementarea unei interfețe ușor de utilizat
- integrarea temelor Light și Dark, cu posibilitatea comutării dinamice
- suport pentru redare audio completă
- gestionarea playlisturilor și a albumelor
- afișarea metadatelor melodiilor
- colectarea și afișarea unor statistici audio
- proiectarea și integrarea unui egalizator audio
- posibilitatea descărcării melodiilor din linkuri
- integrarea unui sistem de favorite
- vizualizarea istoricului melodiilor ascultate
- asigurarea unei arhitecturi modulare, care să permită extinderea și menținerea aplicației

2. Prezentarea aplicației

Aplicația are implementată un media player funcțional:

- redare audio și pauză
- navigare între melodii prin butoanele *Next* și *Previous*
- slider funcțional pentru afișarea și controlul progresului melodiei
- ajustarea volumului
- utilizarea modurilor de redare *Shuffle* și *Repeat*
- selectarea unei melodii direct din listă
- afișarea titlului melodiei aflate în redare
- gestionarea playlisturilor și a albumelor audio
- afișarea metadatelor melodiilor (titlu, artist, album etc.)
- salvarea și gestionarea melodiilor favorite
- posibilitatea descărcării melodiilor din linkuri externe
- comutarea dinamică între temele vizuale Light și Dark

Interfața **Light Theme** include următoarele componente:

- Top Toolbar;
- Media Toolbar;
- meniu lateral;
- listă stilizată de melodii, cu elemente personalizate de tip SongItem.

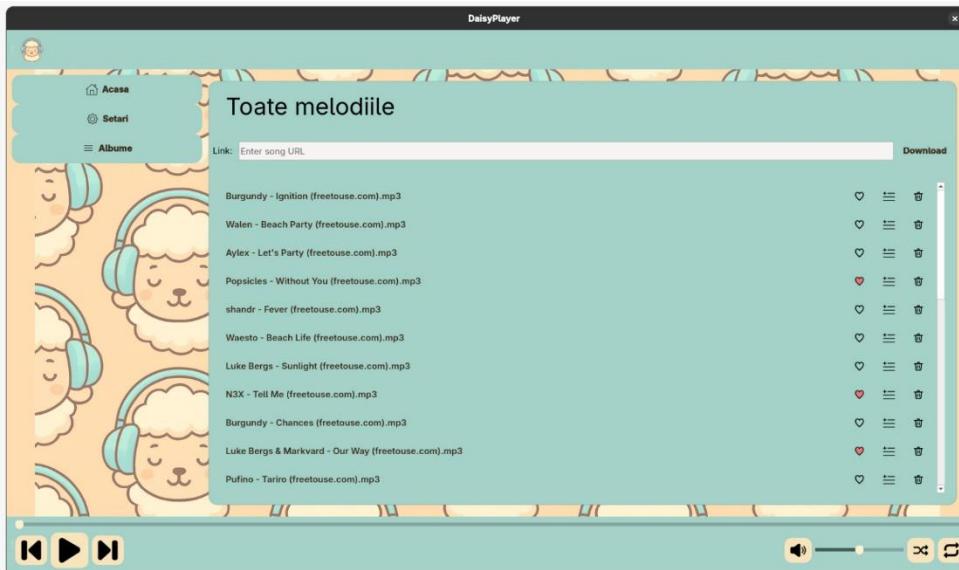


Fig. 1. Ilustrarea temei light în aplicația DaisyPlayer

Interfața **Dark Theme** permite schimbarea schemei de culori prin selectarea opțiunii corespunzătoare din meniul de setări, oferind:

- modificarea culorii fundalului;
- schimbarea culorii textului;
- adaptarea culorilor butoanelor;
- personalizarea elementelor de tip scroll.

Această temă are rolul de a îmbunătăți experiența de utilizare și de a reduce oboseala ochilor în condiții de utilizare îndelungată.

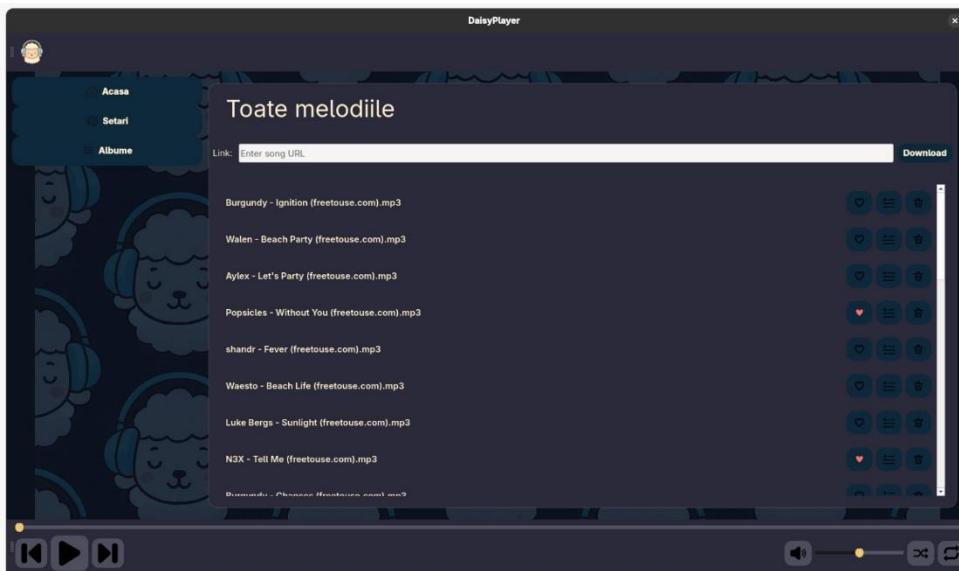


Fig. 3. Ilustrarea temei dark în aplicația DaisyPlayer

3. Descrierea tehnică a aplicației

3.1 Arhitectura aplicației

Aplicația DaisyPlayer este realizată folosind o arhitectură modulară, care permite separarea clară a responsabilităților și facilitează extinderea și menținerea aplicației. Structura proiectului este organizată pe mai multe componente independente, fiecare având un rol bine definit în funcționarea aplicației.

Componenta PlayerService este responsabilă de gestionarea redării audio, folosind clasele QMediaPlayer și QAudioOutput, asigurând funcționalitățile de bază precum redarea, pauza și controlul fluxului audio. Componenta Controller acționează ca un intermediar între interfața grafică și backend, coordonând interacțiunea dintre utilizator și logica aplicației. Interfața grafică (UI) este separată de logica aplicației, fiind responsabilă exclusiv de afișarea elementelor vizuale și de captarea acțiunilor utilizatorului.

Această abordare arhitecturală contribuie la o mai bună organizare a codului, permite reutilizarea componentelor și face posibilă adăugarea de funcționalități noi fără a afecta structura existentă a aplicației.

3.2 Backend – PlayerService

Componenta PlayerService reprezintă nucleul backend al aplicației DaisyPlayer și este responsabilă de gestionarea tuturor operațiilor legate de redarea audio. Aceasta utilizează clasele QMediaPlayer și QAudioOutput din cadrul modulului QtMultimedia pentru a asigura redarea fișierelor audio într-un mod stabil și eficient.

Prin intermediuclasei QMediaPlayer sunt implementate funcționalitățile de bază ale media playerului, precum redarea, pauza, oprirea și navigarea între melodii. QAudioOutput este utilizat pentru controlul volumului și pentru direcționarea fluxului audio către dispozitivul de ieșire al sistemului. De asemenea, PlayerService gestionează starea redării curente, permitând sincronizarea corectă a progresului melodiei cu elementele vizuale din interfață, precum sliderul de progres.

Logica redării audio este centralizată în această componentă, ceea ce permite separarea clară a funcționalităților de procesare audio de interfața grafică. Această abordare contribuie la o structură clară a aplicației și facilitează eventualele optimizări sau extinderi ale funcționalităților audio în viitor.

3.3 Controller

Componenta Controller are rolul de a coordona interacțiunea dintre interfața grafică și componenta backend a aplicației. Aceasta acționează ca un strat de legătură între acțiunile utilizatorului și logica de redare audio implementată în PlayerService.

Controller-ul gestionează evenimentele generate de interfața grafică, precum apăsarea butoanelor de redare, pauză, navigare între melodii, modificarea volumului sau selectarea unei melodii din listă. În urma acestor evenimente, Controller-ul apelează metodele corespunzătoare din PlayerService și actualizează starea aplicației.

Prin centralizarea gestionării evenimentelor, componenta Controller contribuie la separarea clară a responsabilităților dintre interfața grafică și logica aplicației, asigurând o structură organizată și ușor de întreținut.

3.4 Interfață grafică (UI)

Interfața grafică a aplicației DaisyPlayer este realizată utilizând biblioteca PySide6, fiind organizată în componente distincte, fiecare având un rol bine definit. Această organizare permite o structurare clară a codului și o integrare eficientă cu logica aplicației.

Componentele principale ale interfeței grafice sunt:

- MainWindow, care reprezintă fereastra principală a aplicației și gestionează layout-ul general, precum și integrarea tuturor componentelor vizuale;
- TopToolbar, utilizată pentru afișarea controalelor și informațiilor principale ale aplicației;
- MediaToolbar, care conține comenzi de redare audio, precum play, pauză, navigare între melodii și controlul volumului;
- LeftMenu, care oferă utilizatorului opțiuni de navigare și acces rapid la funcționalitățile aplicației;
- RightMenu, utilizată pentru afișarea informațiilor suplimentare și a opțiunilor de configurare;
- SongItem, componentă personalizată folosită pentru reprezentarea individuală a fiecărei melodii din listă, permitând selectarea și interacțiunea cu fișierele audio.

Stilizarea interfeței grafice este realizată prin intermediul fișierelor QSS (Qt Style Sheets), care permit definirea separată a aspectului vizual față de logica aplicației. Utilizarea fișierelor QSS facilitează modificarea și extinderea temelor vizuale, precum și menținerea unui cod curat și bine organizat. Din punct de vedere tehnic, această abordare contribuie la separarea responsabilităților și la o mai bună menenanță a aplicației.

4. Modul de utilizare al aplicației

Aplicația DaisyPlayer a fost concepută pentru a fi ușor de utilizat, oferind utilizatorului acces rapid la toate funcționalitățile principale ale unui media player. Interacțiunea cu aplicația se realizează exclusiv prin intermediul interfeței grafice.

4.1 Pornirea aplicației

- aplicația se pornește prin rularea fișierului main.py;
- la lansare, este afișată fereastra principală a aplicației, care conține lista de melodii, meniu lateral și comenzi de redare.

4.2 Încărcarea și selectarea melodiei

- fișierele audio sunt plasate în directorul local_music/;
- melodii disponibile sunt afișate automat în lista de redare;
- utilizatorul poate selecta o melodie printr-un click direct pe elementul corespunzător din listă.

4.3 Controlul redării audio

Utilizatorul poate controla redarea audio folosind comenzi disponibile în Media Toolbar:

- pornirea și oprirea redării (play/pause);
- navigarea între melodii utilizând butoanele Next și Previous;
- afișarea în timp real a titlului melodiei aflate în redare.

4.4 Controlul progresului și al volumului

- progresul melodiei este afișat prin intermediul unui slider, care indică poziția curentă în cadrul piesei;
- utilizatorul poate modifica poziția de redare prin deplasarea sliderului;
- volumul sunetului poate fi ajustat folosind controlul dedicat din interfață.

4.5 Moduri de redare

Aplicația oferă mai multe moduri de redare pentru a îmbunătăți experiența utilizatorului:

- activarea modului Shuffle pentru redarea melodiilor într-o ordine aleatorie;
- activarea modului Repeat pentru repetarea melodiei curente sau a listei de redare.

4.6 Funcționalități suplimentare

- utilizatorul poate marca melodii ca favorite, acestea fiind salvate și accesibile ulterior;
- aplicația permite afișarea metadatelor melodiilor, precum titlul, artistul și albumul;
- aplicația oferă posibilitatea descărcării melodiilor din linkuri externe;
- utilizatorul poate comuta dinamic între temele vizuale Light și Dark din meniul de setări.

5. Contribuția membrilor echipei

Posmangiu Silviu-Andrei – Backend & Logică aplicație

- implementarea logicii de redare audio a aplicației
- integrarea modulelor QMediaPlayer și QAudioOutput pentru gestionarea fluxului audio
- dezvoltarea funcționalităților de bază ale media playerului (play, pause, next, previous)
- implementarea și sincronizarea sliderului de progres al melodiei
- gestionarea metadatelor melodiilor
- implementarea modurilor de redare Shuffle și Repeat
- colaborarea la integrarea backend-ului cu interfața grafică

Ghiță Mihaela-Gabriela – UI, Design & Documentație

- proiectarea și implementarea interfeței grafice a aplicației folosind PySide6
- realizarea și integrarea temelor vizuale Light Theme și Dark Theme, cu posibilitatea comutării dinamice
- stilizarea interfeței grafice utilizând fișiere QSS (Qt Style Sheets)
- crearea și personalizarea componentelor vizuale (SongItem, meniuri, layout)
- integrarea coerentă dintre interfața grafică și logica aplicației
- realizarea documentației proiectului, incluzând structurarea, redactarea și actualizarea conținutului final al raportului

6. Dependențele aplicației

Pentru dezvoltarea și rularea aplicației DaisyPlayer au fost utilizate următoarele tehnologii și biblioteci software:

- **Python 3.10** – limbajul de programare principal utilizat pentru implementarea aplicației
- **PySide6 (Qt for Python 6.6)** – bibliotecă utilizată pentru realizarea interfeței grafice
- **Biblioteci standard Python** – module precum os și sys sunt utilizate pentru gestionarea fișierelor, a căilor de acces și a resurselor aplicației
- **Module Qt** – aplicația utilizează următoarele module din cadrul PySide6:

- **QtWidgets**, pentru crearea și gestionarea componentelor grafice
- **QtGui**, pentru gestionarea elementelor grafice și a stilizării
- **QtCore**, pentru funcționalități de bază precum semnale, sloturi și gestionarea evenimentelor
- **QtMultimedia**, pentru redarea fișierelor audio și controlul fluxului multimedia
- **yt-dlp**, pentru descărcarea melodii de pe internet
- **mutagen**, pentru calcularea timpului de redare al unui album

7. Modul de execuție

Pentru rularea aplicației DaisyPlayer este necesară instalarea limbajului Python 3.10, precum și a bibliotecilor din “requirements.txt”.

Instalarea dependențelor se realizează din terminal prin comanda: “pip install -r requirements.txt”

Aplicația se pornește prin rularea fișierului principal: “python main.py”

Fișierele audio utilizate de aplicație trebuie adăugate în directorul “local_music/” din structura proiectului. Melodiile plasate în acest director sunt încărcate automat și afișate în lista de redare a aplicației.

8. Concluzii și direcții viitoare

În cadrul acestui proiect a fost dezvoltată aplicația DaisyPlayer, un media player funcțional realizat în Python folosind biblioteca PySide6. Aplicația îndeplinește obiectivele propuse, oferind o interfață grafică intuitivă, suport complet pentru redarea fișierelor audio și funcționalități moderne de gestionare a melodii.

Ca direcții de dezvoltare viitoare, aplicația poate fi extinsă prin adăugarea de funcționalități suplimentare, precum suport pentru streaming online și gestionarea avansată a playlisturilor.