
Documentul Specificațiilor Cerințelor

pentru

eMusic Player

**Realizat de Anghel Ioana,
Erich Lemny,
Popescu Vlad,
Anastasei Narcis,
grupa 1311A**

Universitatea Tehnică „Gheorghe Asachi”

27.05.2024

Cuprins

Cuprins	ii
Istoricul reviziilor	ii
1. Introducere	1
1.1 Scop	1
1.2 Convențiile documentului	1
1.3 Audiența țintă și sugestii de citire	1
1.4 Aria produsului	1
1.5 Referințe	1
2. Descrierea generală	2
2.1 Perspectiva produsului	2
2.2 Funcțiile produsului	2
2.3 Clasele și caracteristici	2
2.4 Mediul de operare	2
2.5 Constrângeri de implementare și design	2
2.6 Documentația	2
2.7 Asumții și dependențe	
3. Cerințele interfețelor externe	3
3.1 Interfețele cu utilizatorul	3
3.2 Interfețele hardware	3
3.3 Interfețele software	3
3.4 Interfețele de comunicare	3
4. Trăsături de sistem	4
4.1 Redarea unui cântec	4
4.2 Încărcarea și salvarea unui playlist	4
4.3 Editarea tag-urilor unui cântec	4
5. Alte cerințe non-funcționale	4
5.1 Cerințe de performanță	4
5.2 Cerințe de siguranță	5
5.3 Cerințe de securitate	5
5.4 Atribute ale calității	5
5.5 Reguli de business	5

1. Introducere

1.1 Scop

Scopul principal al aplicației “eMusic Player” este de a reda fișiere audio pe orice calculator, într-un mod rapid și intuitiv pentru orice utilizator. Fișierele acceptate de acest player nu sunt doar în format mp3, utilizatorul putând alege de pe computerul propriu unul sau mai multe fișiere pentru a le putea asculta.

Playerul are funcționalități de bază, cu mențiunea ca utilizatorul își poate crea mai multe playlisturi și ulterior le poate căuta după cuvinte cheie. De asemenea, pot fi modificate anumite detalii pentru fiecare piesă din playlist. În plus, programul are un help asociat pentru a ajuta utilizatorul să se orienteze mai ușor cu privire la anumite aspecte.

1.2 Convențiile documentului

Acest document urmează anumite convenții și standarde tipografice pentru a asigura claritatea și consistența prezentării Specificațiilor Cerințelor Software (SRS) pentru o aplicație simplă de music player offline. Aceste convenții sunt descrise mai jos:

- Titluri: Titlurile principale (nivel 1) sunt scrise cu font Times New Roman, Bold, mărimea 14. Subtitlurile (nivel 2) sunt scrise cu font Times New Roman, Bold, mărimea 12. Titlurile de nivel 3 și nivelurile inferioare folosesc font Times New Roman, Italic, mărimea 12.
- Textul principal: Textul general al documentului este scris cu font Times New Roman, mărimea 12.
- Elemente de cod: Orice fragment de cod sau referință la elemente tehnice este scris cu font Courier New, mărimea 10.
- Cerințele sunt etichetate cu priorități de la 1 la 3, unde 1 indică o prioritate ridicată (cerință critică), 2 indică o prioritate medie (cerință importantă dar nu critică), și 3 indică o prioritate scăzută (cerință opțională).
- Prioritățile pentru cerințele de nivel superior sunt moștenite de cerințele detaliate corespunzătoare, cu excepția cazurilor în care se specifică altfel în mod explicit.
- Bold: Folosit pentru a sublinia cuvinte cheie și concepte importante.
- Italic: Folosit pentru a sublinia titluri de secțiuni și pentru termeni tehnici și definiții.
- Subliniat: Folosit rar, doar pentru a scoate în evidență elemente esențiale care necesită o atenție deosebită.
- Diagramele și ilustrațiile sunt etichetate cu numere secvențiale și descrieri scurte (de exemplu, „Figura 1: Diagrama arhitecturală a playerului de muzică”).
- Toate diagramele sunt realizate folosind standarde UML pentru a asigura claritatea și înțelegerea comună.

- Notele explicative sunt inserate la subsolul paginilor relevante și sunt numerotate secvențial.
- Referințele externe sunt enumerate într-o secțiune separată la finalul documentului și urmează stilul de citare IEEE.

Aceste convenții sunt menite să faciliteze citirea și înțelegerea documentului SRS, asigurând în același timp consistența în întreaga documentație. Toate părțile implicate în dezvoltarea și revizuirea acestui document sunt încurajate să respecte aceste convenții pentru a menține standardele de calitate și claritate.

1.3 Audiența țintă și sugestii de citire

Acest document este destinat unei audiențe diverse implicate în dezvoltarea și utilizarea aplicației de music player offline. Fiecare tip de cititor va găsi secțiuni relevante în acest document, structurate astfel încât să faciliteze accesul rapid la informațiile necesare.

Tipuri de cititori:

- Dezvoltatori: Responsabili pentru implementarea și mentenanța software-ului. Ei vor găsi detalii tehnice și specificații funcționale necesare pentru a construi aplicația.
- Manageri de proiect: Responsabili pentru planificarea și gestionarea proiectului. Vor găsi utilă structura documentului, cerințele și prioritățile pentru a supraveghea progresul și resursele alocate.
- Echipa de marketing: Responsabili pentru promovarea aplicației. Vor folosi descrierile funcționalităților și beneficiilor pentru a elabora strategii de marketing.
- Utilizatori finali: Persoanele care vor utiliza aplicația. Documentul oferă o prezentare generală a funcțiilor și interfeței aplicației.
- Tester: Responsabili pentru verificarea calității și performanței aplicației. Vor folosi criteriile de acceptare și scenariile de testare descrise în document.
- Scriitori de documentație: Responsabili pentru crearea ghidurilor de utilizare și altor materiale de suport. Vor găsi descrierile funcționalităților și instrucțiunile utile.

Sugestii de citire:

- Pentru toți cititorii: Începeți cu secțiunea Introducere pentru a înțelege scopul și structura documentului.
- Manageri de proiect și echipa de marketing: Continuați cu Descriere generală pentru a obține o imagine de ansamblu asupra produsului și a funcționalităților acestuia.
- Dezvoltatori: După citirea secțiunii de Descriere generală, aprofundați secțiunea Cerințe specifice, oferind o atenție sporită cerințelor funcționale și interfețelor utilizatorului.

- **Tester:** Examinați secțiunea Cerințe specifice, în special cerințele de performanță și criteriile de acceptare, pentru a planifica și realiza testele necesare.
- **Scriitori de documentație:** După parcurgerea secțiunii Descriere generală, treceți la Cerințe specifice pentru a aduna informațiile necesare pentru elaborarea documentației de utilizare.
- **Utilizatori finali:** Parcurgeți secțiunea Descriere generală pentru a înțelege funcționalitățile și beneficiile aplicației.

Aceste sugestii de citire sunt menite să optimizeze experiența fiecărui tip de cititor și să faciliteze accesul la informațiile cele mai relevante și utile pentru rolul lor specific în cadrul proiectului.

1.4 Aria produsului

Aplicația este concepută pentru a oferi utilizatorilor posibilitatea de a asculta muzică stocată local pe dispozitivele lor, fără a necesita o conexiune la internet.

Aplicația eMusic Player permite utilizatorilor să:

- Redea melodii stocate local în diferite formate audio (MP3, WAV, FLAC, etc.).
- Creeze, editeze și salveze playlisturi.
- Controleze redarea muzicii (play, pause, stop, prev, next).
- Vizualizeze informații despre melodii (titlu, artist, durată).

Beneficii relevante:

- **Utilizare fără conexiune la internet:** Aplicația funcționează complet offline, oferind utilizatorilor acces la muzică în orice moment și în orice loc, fără a necesita o conexiune la internet.
- **Management eficient al muzicii:** Utilizatorii pot organiza și gestiona eficient colecțiile lor de muzică prin playlisturi.
- **Interfață intuitivă:** Oferă o interfață prietenoasă și ușor de utilizat, potrivită pentru toate categoriile de utilizatori.

Relația cu obiectivele corporative:

Aplicația de music player offline se aliniază cu strategiile de business ale companiei de a oferi soluții software inovatoare și de înaltă calitate care îmbunătățesc experiența utilizatorilor. Acest produs extinde portofoliul de aplicații multimedia ale companiei, contribuind la creșterea satisfacției și fidelizării clienților.

1.5 Referințe

<https://refactoring.guru/design-patterns/state>

<https://refactoring.guru/design-patterns/facade>

<https://github.com/filoe/cscore>

<https://github.com/mono/taglib-sharp>

Cursul si laboratorul de IP.

2. Descrierea generală

2.1 Perspectiva produsului

Aplicația descrisă este o replică a unui player “Winamp”, dezvoltat pentru a satisface nevoile utilizatorilor care doresc să asculte muzică fără a depinde de o conexiune la internet. Acest produs nu este un membru următor al unei familii de produse și nu este o înlocuire pentru sisteme existente.

Aplicația interacționează cu diferite componente și sisteme externe, precum și cu utilizatorii finali. Principalele interfețe includ:

- Sistemul de fișiere al dispozitivului: Pentru accesarea și gestionarea fișierelor muzicale stocate local.
- API-uri de sistem: Pentru funcționalități de redare audio, control al volumului și alte interacțiuni cu hardware-ul dispozitivului.
- Interfața utilizatorului: Pentru a permite utilizatorilor să interacționeze cu aplicația prin intermediul dispozitivelor de input.

//de inserat diagrama arhitecturala

2.2 Funcțiile produsului

Aplicația eMusic Player trebuie să îndeplinească următoarele funcții majore pentru a satisface nevoile utilizatorilor. Detaliile acestor funcții vor fi prezentate în Secțiunea 3.

Funcții principale:

- Creare playlist nou: utilizatorii pot crea playlisturi noi pentru a organiza și gestiona colecțiile lor de muzică.
- Adăugare melodie în playlist din memoria locală: utilizatorii pot adăuga melodii stocate local pe dispozitiv în playlisturile existente.
- Redare playlist: funcții de control al redării muzicii, inclusiv Start (începerea redării unui playlist), Stop (oprirea redării), Pauză (pauzarea redării), Previous (redarea melodiei anterioare), Next (redarea melodiei următoare).
- Adăugare sau editare detalii playlist: utilizatorii pot adăuga sau edita detalii despre playlisturi, inclusiv an, nume, titlu, album, gen.
- Salvare playlist: utilizatorii pot salva playlisturile create sau modificate pentru acces ulterior.

2.3 Clasele și caracteristici

Pentru aplicația de music player offline, se anticipează următoarele clase de utilizatori, fiecare având caracteristici și nevoi distincte:

Clasele de utilizatori:

1. Utilizatori ocazionali:

- Frecvența utilizării: Utilizare rară sau ocazională.
- Subsetul de funcții utilizate: Redare de bază a muzicii (start, stop, pauză, next, previous).
- Expertiză tehnică: Scăzută până la moderată.
- Nivel de securitate sau privilegii: Standard.
- Nivel educațional sau experiență: Variază, dar în general, fără necesitatea de a avea cunoștințe tehnice avansate.

2. Utilizatori frecvenți:

- Frecvența utilizării: Utilizare zilnică sau frecventă.
- Subsetul de funcții utilizate: Redare avansată a muzicii, gestionarea playlisturilor, personalizarea interfeței.
- Expertiză tehnică: Moderată până la avansată.
- Nivel de securitate sau privilegii: Standard.
- Nivel educațional sau experiență: Utilizatori cu ceva experiență în utilizarea aplicațiilor muzicale și tehnologiei.

Caracteristicile utilizatorilor

1. Utilizatori ocazionali:

- Au nevoie de o interfață simplă și intuitivă.
- Utilizează funcții de bază, cum ar fi redarea și gestionarea simplă a playlisturilor.
- Nu sunt interesați de funcții avansate sau personalizare detaliată.

2. Utilizatori frecvenți:

- Caută o experiență de utilizare fluidă și funcționalități suplimentare, cum ar fi crearea și salvarea playlisturilor.
- Apreciază opțiunile de personalizare și detaliile suplimentare ale melodiilor.
- Au nevoie de o interfață ușor de navigat, dar cu acces la funcții mai complexe.

Această clasificare a utilizatorilor permite o înțelegere clară a cerințelor și nevoilor fiecărei clase, asigurând că aplicația este proiectată pentru a satisface diversele așteptări și niveluri de expertiză ale utilizatorilor.

2.4 Mediul de operare

Aplicația eMusic Player offline este concepută pentru a funcționa într-un mediu local, având cerințe minime de resurse.

Detaliile despre mediul de operare sunt prezentate mai jos:

1. Platformă hardware:

- Dispozitive: Computere personale, laptopuri, tablete și smartphone-uri.
- Configurație minimă recomandată:
Procesor: Dual-core, 1.5 GHz sau mai mare

RAM: 2 GB sau mai mult

Stocare: spațiu pentru fișierele muzicale

2. Sisteme de operare

Aplicația este compatibilă cu principalele sisteme de operare, inclusiv:

- Windows: Windows 10 și versiuni mai noi

3. Alte componente software

- Player Audio: Aplicația utilizează API-urile native ale sistemului de operare pentru redarea audio.
- Acces la sistemul de fișiere: Aplicația necesită permisiuni pentru a accesa și gestiona fișierele stocate local.
- Interfața grafică: Biblioteci grafice native pentru fiecare sistem de operare.

4. Rețea și conectivitate

Fără necesitate de conexiune la internet: Aplicația este concepută pentru utilizare offline și nu necesită o conexiune la internet pentru funcțiile sale principale.

2.5 Constrângeri de implementare și design

Una din probleme întâmpinate este cea a afișare a melodiei curente: nu putem afișa titlul acesteia deoarece este posibil ca fișierul mp3 să nu aibă tag-ul de artist setat și, dacă dintr-un playlist sunt mai multe melodii, e mai bine să afișăm cel puțin numele fișierului pentru a diferența între melodii, în loc de afișarea unui text placeholder.

De asemenea, nu recomandăm rularea a doua sau mai multe instanțe ale aplicației, deoarece există posibilitatea ca sunetul să nu fie redat corespunzător, sau sunetul să continue să se audă chiar dacă o instanță se închide.

2.6 Documentația

Aplicația conține o secțiune de "Help". Aceasta este destinată utilizatorului și reprezintă un ghid în folosirea eMusic Player.

În cadrul secțiunii de "Help" regăsim:

- o secțiune de introducere care surprinde două pagini: o pagină destinată prezentării aplicației, iar cealaltă îndrumă utilizatorul despre ce facilități dispune aplicația.
- o secțiune „Getting Started” cu rolul de a prezenta specificațiile tehnice pentru o bună funcționare a aplicației.
- o secțiune de prezentare a interfeței cu utilizatorul, care conține alte trei pagini diferite în care se prezintă modul în care pot fi încărcate melodiile, modul de salvare a acestora și cum se poate personaliza un playlist.

2.7 Asumpții și dependențe

Din diverse motive, componenta de editare a informațiilor despre melodii (cunoscute drept tag-uri) are niste limitari. De aceea, fiecare melodie are următoarele informații:

- O melodie poate contine doar UN artist
- O melodie poate aparține unui SINGUR gen muzical

Libraria folosită pentru implementarea funcțiilor de redare a sunetului depind de modul în care Windows o face, orice modificare semnificativă la nivelul intern al sistemului de operare poate cauza probleme pe viitor.

3. Cerințele interfețelor externe

3.1 Interfețele cu utilizatorul

Principala interfata este intuitiva si se poate adresa oricarui utilizator. La pornirea aplicației se deschide o fereastră ce contine cateva butoane de control de baza. La prima deschidere a player-ului pe un computer nu exista piese in playlist, astfel ca, dacă utilizatorul va apăsa butonul de Play, se va afișa mesajul “No songs to play!”, acesta reprezentand că trebuie incarcat măcar un fișier audio pentru a functiona acest buton. Se pot încărca mai multe melodii si ulterior poate fi făcut un playlist (sau mai multe) cu acestea, cu posibilitatea de a le salva si de a le putea vizualiza la următoarele deschideri ale aplicației. Exista printre altele si optiunea de a repeta piesa curenta, însă una din cele mai utile opțiuni este căutarea unui playlist dupa niște cuvinte cheie.

3.2 Interfețele hardware

Resursele hardware necesare pentru rularea acestei aplicației sunt unele de baza, utilizatorul avand nevoie de mouse (pentru a apăsa pe butoanele de control, inclusiv pentru a accesa help-ul), tastatura (pentru a cauta piese si playlisturi dupa anumite cuvinte cheie, dar si pentru a edita anumite informații despre piesa curentă) si un monitor/display. Nu exista cerinte minime pentru aceste resurse.

3.3 Interfețele software

Principala resursa de care are nevoie utilizatorul pentru a putea utiliza aplicatia este sistemul de operare Windows, cu mentiunea ca trebuie sa aiba instalat framework-ul .NET, versiunea minima acceptata fiind 4.8.

3.4 Interfețele de comunicare

Aplicația eMusic Player nu necesita conexiune la internet sau mail, intrucat este suficient ca utilizatorul sa aiba in computerul propriu fisiere audio in unul din formatele specificate mai sus

pentru a le putea asculta. Playerul ruleaza local si in aceste conditii piesele si playlist-urile pot ramane salvate pe computerul propriu.

4. Trăsături de sistem

4.1 Redarea unui cântec

4.1.1 Descrierea și prioritatea

Această trăsătură a sistemului se referă la abilitatea aplicației de a porni, opri și a pune pe pauză un cântec. Este de prioritate ÎNALTĂ.

4.1.2 Secvența stimul/răspuns

1. Utilizatorul deschide aplicația.
2. Apasă pe butonul Select Song.
3. Apare o fereastră pentru selectarea unui cântec.
4. Apasă pe butonul Play.
5. Cântecul se pornește.
6. Apasă pe butonul Pause.
7. Cântecul se oprește, dar își salvează poziția.
8. Apasă iar pe Play.
9. Cântecul continuă de unde a rămas.
10. Apasă pe Stop.
11. Cântecul se oprește.
12. Apasă pe Play.
13. Cântecul se pornește de la început.

4.1.3 Cerințe funcționale

Cerința 1: Existența butoanelor Play, Pause, Stop, Select Song.

Cerința 2: Un pop-up când utilizatorul încearcă să dea Play fără să existe un cântec selectat.

Cerința 3: Să existe funcții de callback care vor executa acțiunile corespunzătoare.

4.2 Încărcarea și salvarea unui playlist

4.2.1 Descrierea și prioritatea

Această trăsătură semnifică abilitatea utilizatorului de a încărca în aplicație un playlist deja existent și de a salva unul creat de utilizator. Este de prioritate MEDIE.

4.2.2 Secvență stimul/răspuns

1. Utilizatorul deschide aplicația.
2. Apasă pe butonul Open.
3. Apare o fereastră pentru selectarea unui playlist existent.
4. În cazul inexistenței playlist-ului selectat, aplicația va arunca o excepție.
5. Playlist-ul este încărcat în aplicație.

O altă secvență:

1. Utilizatorul deschide aplicația.
2. Apasă pe Select a song.
3. În fereastra apărută alege un cântec.
4. Repetă pasul 2 de mai multe ori.
5. Apasă pe butonul Save.
6. Se deschide o fereastră pentru salvarea playlist-ului cu un nume introdus de utilizator.
7. Utilizatorul apasă pe Ok.

4.2.3 Cerințe funcționale

Cerința 1: Să existe butoanele Select a song, Save, Open.

Cerința 2: Să existe funcțiile de callback aferente butoanelor.

Cerința 3: Să se afișeze ferestrele necesare pentru Select a song, Save, Open.

Cerința 4: Să se salveze în fișierul creat prin Save căile spre cânticele din playlist.

4.3 Editarea tag-urilor unui cântec

4.3.1 Descrierea și prioritatea

Această trăsătură semnifică abilitatea utilizatorului de a edita tag-urile de tip ID3v2 a unui cântec în format MP3 din aplicație. Este de prioritate MEDIE.

4.3.2 Secvență stimul/răspuns

1. Utilizatorul deschide aplicația.
2. Utilizatorul apasă pe Select a song.
3. Apare fereastra de selectare a unui cântec.
4. După selectare, apare în interfață.
5. Apasă pe butonul Edit Song Info.
6. Apare o altă fereastră cu niște câmpuri text pentru Cântăreț, Album, An, Genuri, Titlu.
7. Utilizatorul schimbă textul din aceste căsuțe.
8. Apasă pe butonul OK.
9. Se salvează tag-urile în fișierul aferent.

4.3.3 Cerințe funcționale

Cerința 1: Să existe butoanele Select a song, Edit Song Info.

Cerința 2: Să existe funcțiile de callback aferente butoanelor.

Cerința 3: Să se afișeze fereastra cu câmpurile text pentru tag-urile Cântăreț, Album, An, Genuri, Titlu.

Cerința 4: Să se salveze la apăsarea butonului OK a tag-urilor modificate.

5. Alte cerințe non-funcționale

5.1 Cerințe de performanță

Singurele cerințe de performanță pentru această aplicație sunt ca la apăsarea butoanelor Play, Pause, Stop acțiunile să se execute într-un timp relativ scurt, sub 0.5 secunde. Plus cântecul trebuie să nu fie întrerupt la redare.

5.2 Cerințe de siguranță

Singura cerință de siguranță este ca fișierul playlist să fie creat corect, cu căile aferente cântecelor selectate. Aceste căi nu trebuie să fie corupte. Totuși nu sunt necesare orice cerințe specifice pentru siguranță.

5.3 Cerințe de securitate

Nu există cerințe de securitate pentru această aplicație, cum ea nu stochează nici un fel de informație după execuție, în afară de cea creată de utilizator, plus nu are acces la Internet. Lucrează doar cu fișiere sonore, nu și de orice altă natură. Informațiile create de utilizator prin playlist-uri nu conțin date de caracter personal, de aceea nu este necesară o oarecare criptare a lor sau alte acțiuni pentru securizarea lor.

5.4 Atribute ale calității

Calități adiționale ale acestui player audio: necesar de memorie scăzut (sub 50 de Mbytes de memorie RAM necesară), nu încarcă procesorul cu calcule multe, doar funcționalități minimale (redarea unui cântec, crearea playlist-urilor, modificarea tag-urilor), des utilizate.

5.5 Reguli de business

Player-ul suportă doar un rol, cel al unui utilizator. Nu există roluri cu funcționalități specializate.

Anexa.1

```

public class Player
{
    private AbstractPlayerState _state;

    public enum States
    {
        NoSongs,
        HasSongs,
        PlayingSong,
        PausedSong
    }

    private States _currentState;
    public States CurrentState
    {
        get
        {
            return _currentState;
        }
        set
        {
            _currentState = value;
        }
    }

    private Song _currentSong;
    private int _currentSongIndex;
    private bool _repeatSong;

    private ListBox _listBoxSongs;
    private Timer _timerPosition;
    private TextBox _currentSongState;

    private List<Song> _allSongs;

    public AbstractPlayerState State
    {
        set
        {
            _state = value;
        }
    }

    public Song CurrentSong
    {
        get
        {
            return _currentSong;
        }

        set
        {
            _currentSong = value;
        }
    }
}

```

```

    public Player(ListBox listBoxSongs, Timer timerPosition, TextBox
currentSongState)
    {
        _state = new NoSongsState(this);
        _allSongs = new List<Song>();
        _listBoxSongs = listBoxSongs;
        _timerPosition = timerPosition;
        _currentSongState = currentSongState;
        _repeatSong = false;
    }

    public string GetCurrentSongName()
    {
        return _listBoxSongs.SelectedItem.ToString();
    }

    public void SetCurrentSongState(string state)
    {
        _currentSongState.Text = state;
    }

    public void SetTimer(bool state)
    {
        _timerPosition.Enabled = state;
    }

    public bool RepeatSong
    {
        get
        {
            return _repeatSong;
        }
        set
        {
            _repeatSong = value;
        }
    }

    public List<string> GetAllSongsLocations()
    {
        List<string> allSongs = new List<string>();

        for (int i = 0; i < _allSongs.Count; i++)
        {
            allSongs.Add(_allSongs[i].Path);
        }

        return allSongs;
    }

    public bool IsEmpty()
    {
        return (_allSongs.Count == 0);
    }

    public void AddSong(Song song)

```

```

{
    _allSongs.Add(song);
}

public void DeleteSongs()
{
    if (_allSongs.Count > 0)
    {
        _allSongs.Clear();
        _currentSongIndex = -1;
        _currentSong = null;
    }
}

public void SelectSong(int index)
{
    _currentSong = _allSongs[index];
    _currentSongIndex = index;
}

public void NextSong()
{
    if (_currentState == States.NoSongs)
        return;

    if (_currentSongIndex == _allSongs.Count - 1)
    {
        _currentSong = _allSongs[0];
        _currentSongIndex = 0;
        _listBoxSongs.SelectedIndex = _currentSongIndex;
    }
    else
    {
        _currentSong = _allSongs[_currentSongIndex + 1];
        _currentSongIndex++;
        _listBoxSongs.SelectedIndex = _currentSongIndex;
    }
}

public void PrevSong()
{
    if (_currentState == States.NoSongs)
        return;

    if (_currentSongIndex == 0)
    {
        _currentSong = _allSongs[_allSongs.Count - 1];
        _currentSongIndex = _allSongs.Count - 1;
        _listBoxSongs.SelectedIndex = _currentSongIndex;
    }
    else
    {
        _currentSong = _allSongs[_currentSongIndex - 1];
        _currentSongIndex--;
        _listBoxSongs.SelectedIndex = _currentSongIndex;
    }
}

```



```
public int CurrentSongIndex
{
    get
    {
        return _currentSongIndex;
    }
    set
    {
        _currentSongIndex = value;
    }
}

public void Play()
{
    _state.Play();
}

public void Pause()
{
    _state.Pause();
}

public void Stop()
{
    _state.Stop();
}
}
```

Generated by UModel

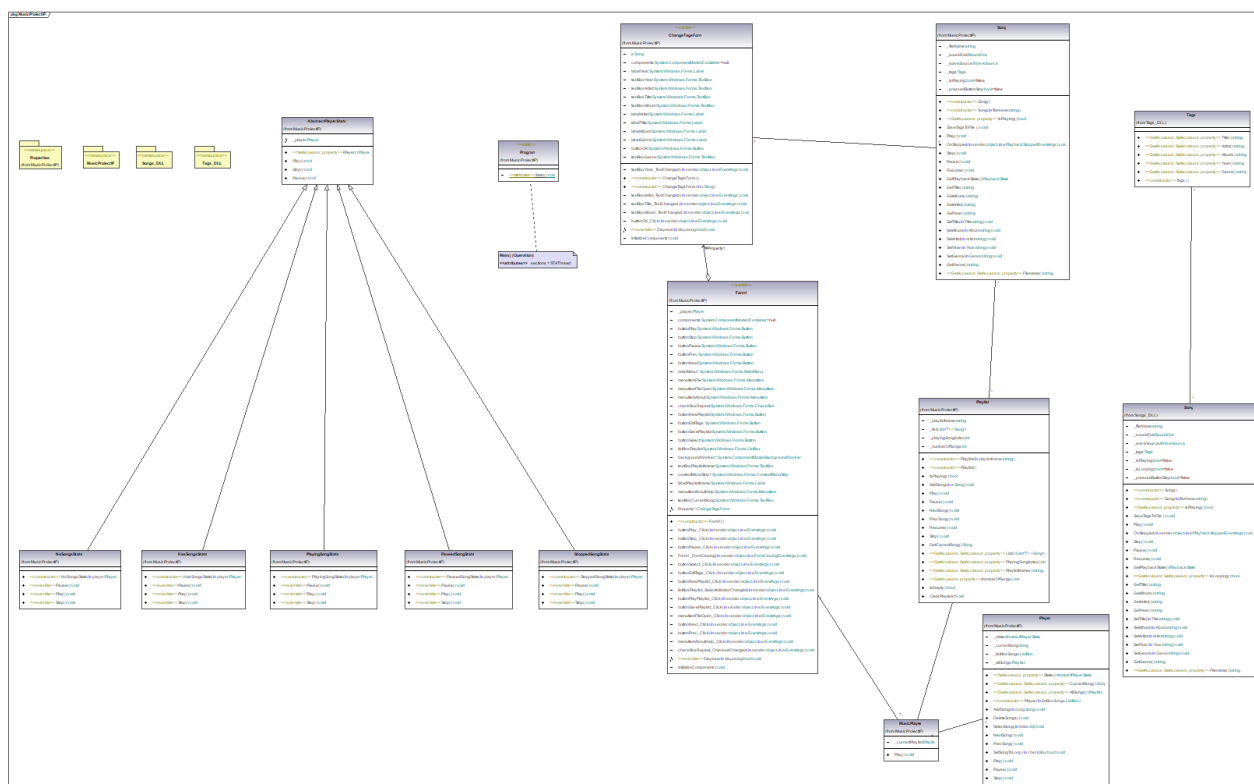


Diagrama de secvențe

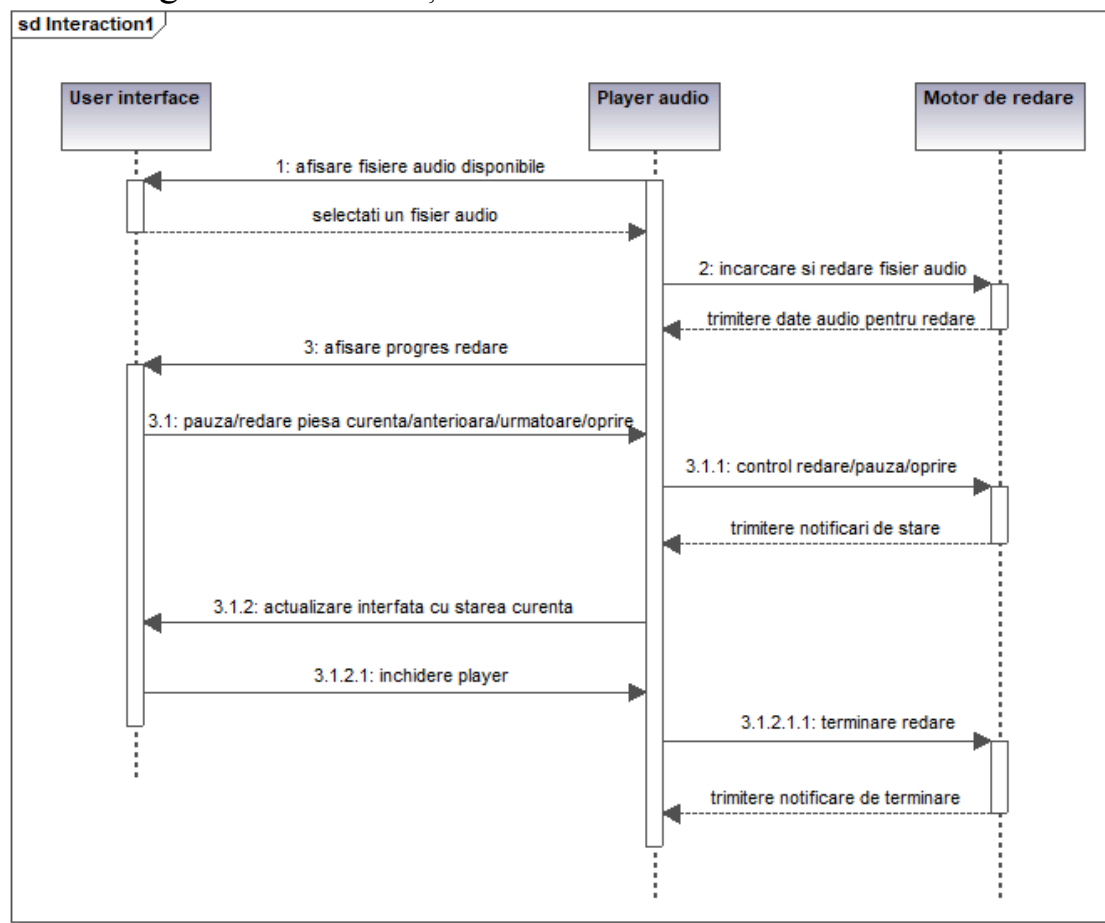


Diagrama de activități

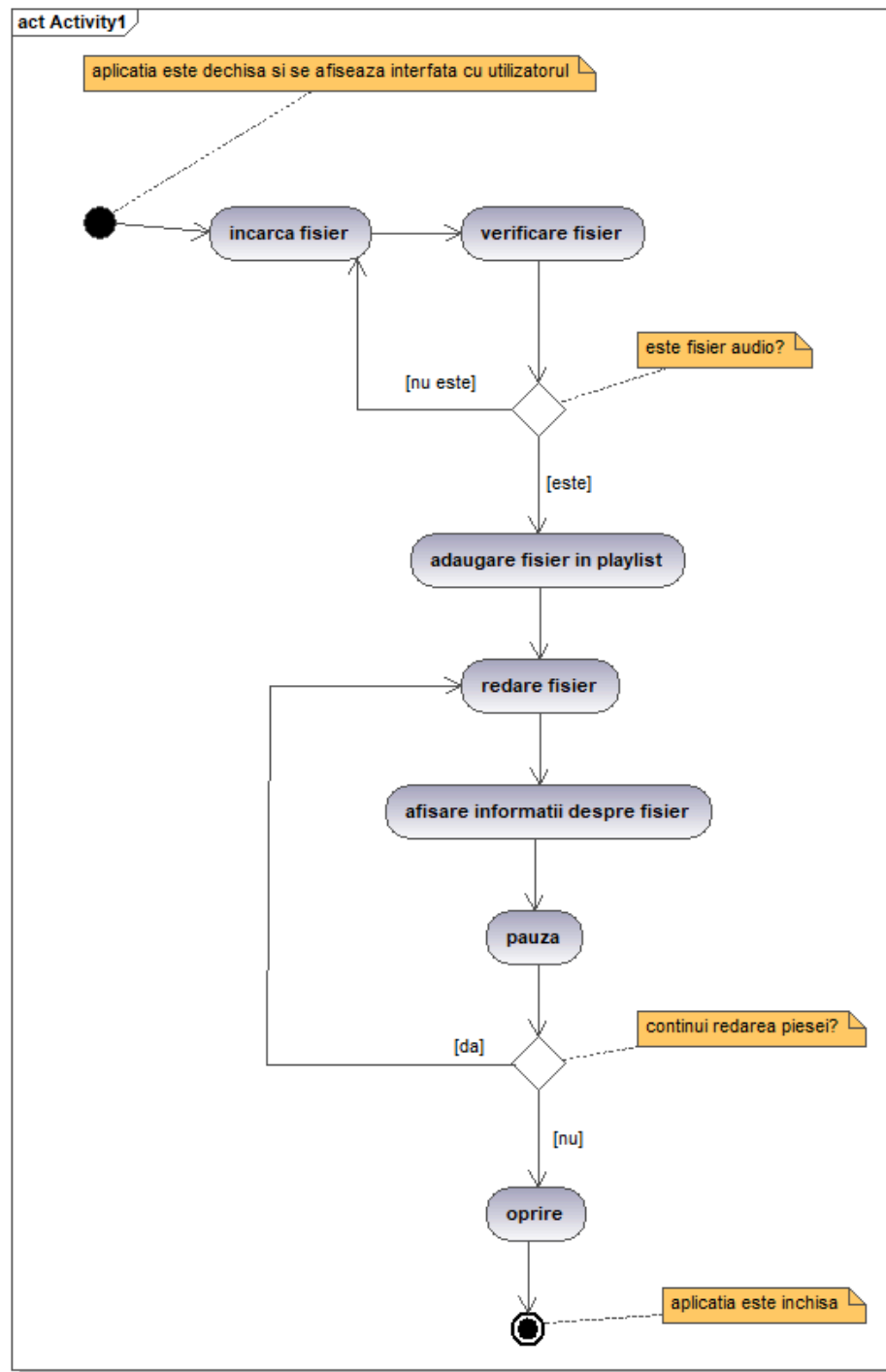
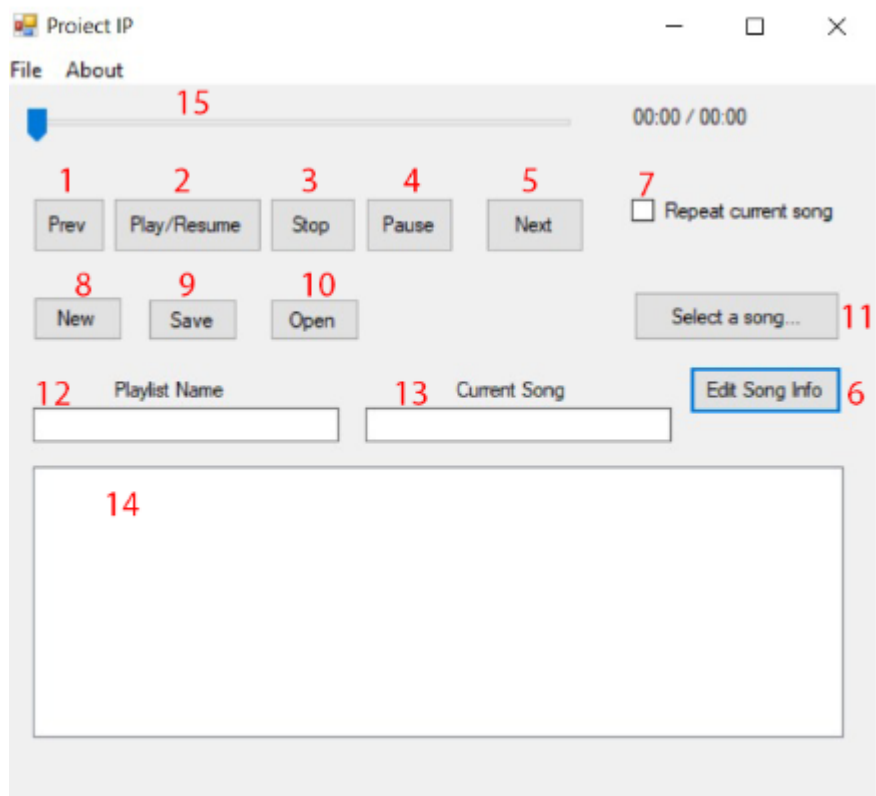


Diagrama de UseCase



Ghid de utilizare a player-ului

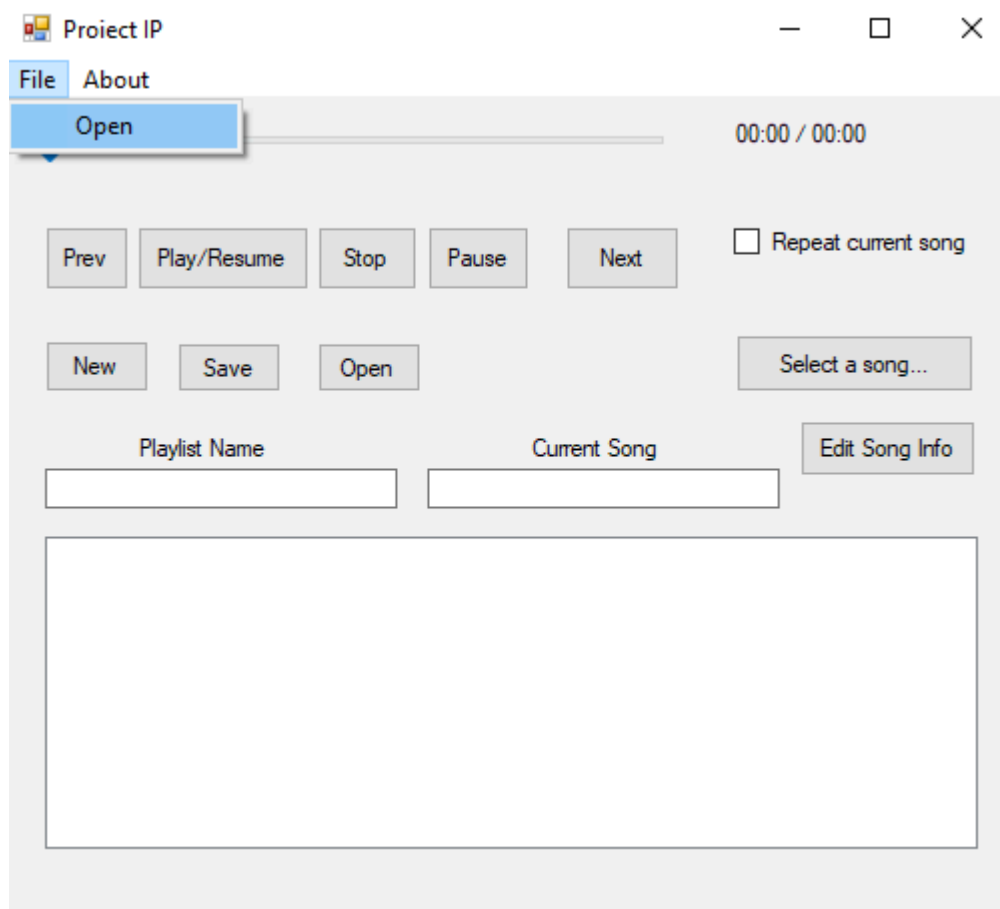
La deschiderea aplicației, se afișează următoarea interfață:



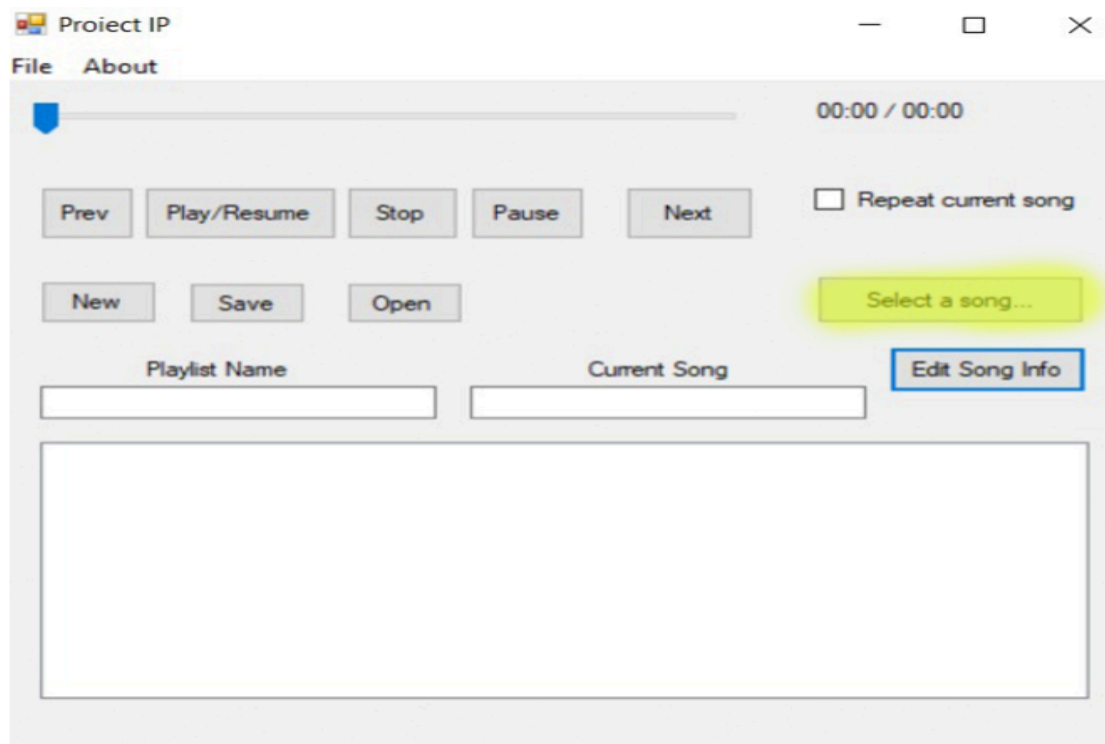
Legendă:

1. Selectează melodia anterioara
2. Redă melodia
3. Oprește redarea
4. Pauză
5. Redă melodia următoare
6. Editează nume artist, nume album, an, gen.
7. Redă melodia actuală la infinit.
8. Playlist nou
9. Salvează playlist-ul
10. Selectează playlist-ul din calculator
11. Selectează cântecul din calculator
12. Numele playlistului curent
13. Numele melodiei curente
14. Lista melodiilor din playlist-ul actual
15. Bara de redare a melodiei curente

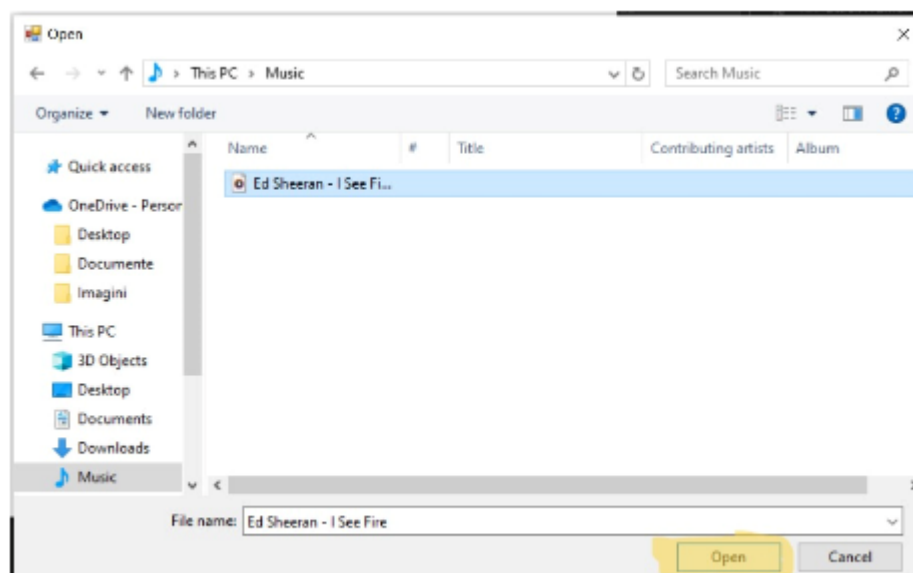
Dacă nu există nicio piesă în playlist, se încarcă un fișier audio astfel: File → Open → se deschide o fereastră



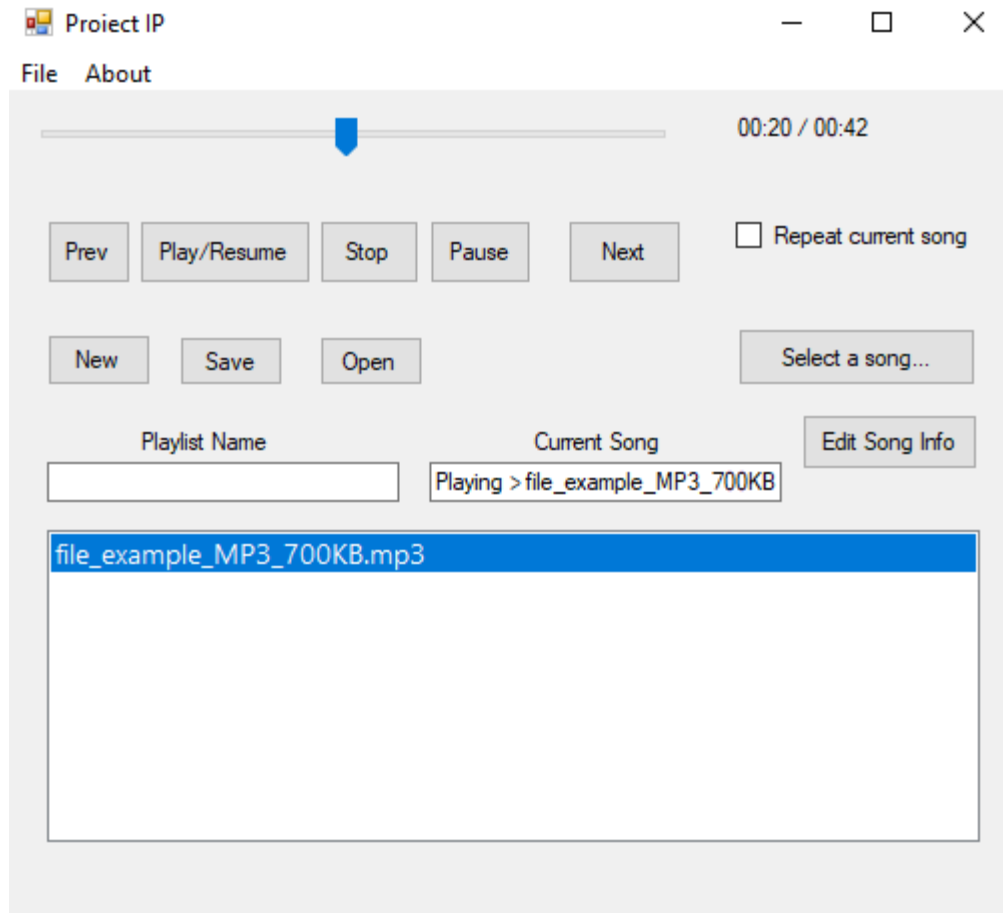
sau



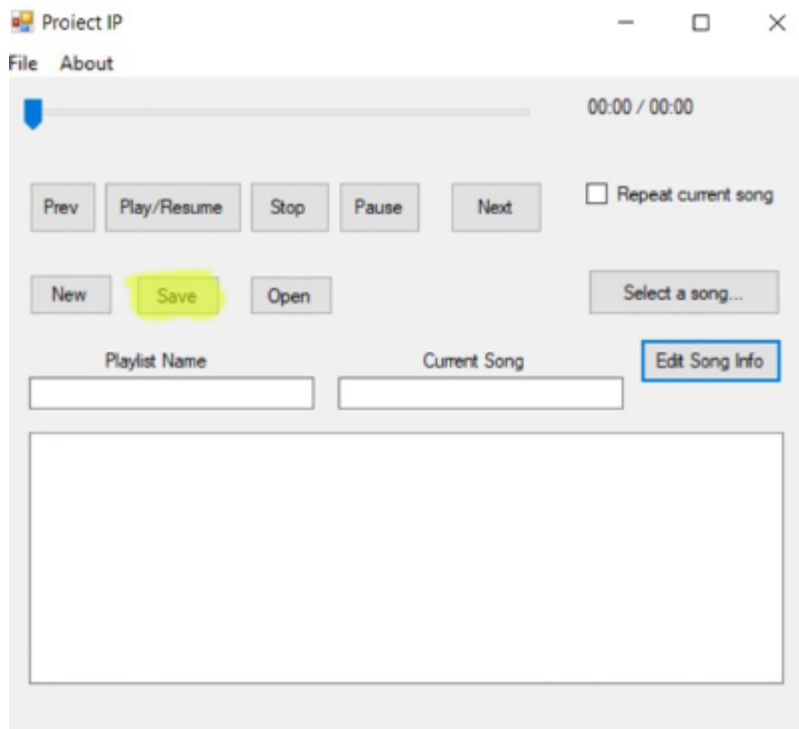
După ce am selectat butonul se va deschide o fereastră care ne ajută să găsim locația la care este salvat cântecul. Selectăm cântecul pe care dorim să îl adăugăm.



Apăsăm butonul "Open". Observăm că melodia selectată a apărut în fereastră.



Pentru a putea salva playlist-ul, putem crea un fișier .txt în care să încărcăm conținutul playlistului. Vom apăsa butonul "Save" de pe interfață.



După ce am apăsut butonul save ni se va deschide o fereastră în care selectăm locația unde dorim sa fie salvat fișierul (1), cum dorim să se numească fișierul (2) și în final să salvăm modificările făcute (3).

