

# TuneWave

Aplicația ta de redare MP3

Realizat de:

Dancău Rareș-Andrei, 1308A

Notă: Acesta este un proiect separat de proiectul pe care-l fac în echipă.

# Cuprins

1. Scopul documentului.....	4
2. Prezentare generală a documentului.....	4
3. Descriere generală a produsului.....	4
3.1. Situația curentă.....	4
3.2. Scopul produsului.....	5
3.3. Contextul produsului.....	5
3.4 Beneficii.....	5
4. Cerinte funcționale.....	6
4.1. Actori.....	6
4.2. Cazuri de utilizare.....	6
4.2.1. Redarea unui fișier mp3 individual.....	6
4.2.2. Redarea unui playlist.....	7
4.2.3. Redarea unui post radio.....	8
4.2.4. Crearea unui playlist.....	8
4.2.5. Modificarea unui playlist existent.....	9
5. Cerinte nefuncționale.....	10
5.1. Cerințe pentru interfața cu utilizatorul.....	10
5.2. Cerințe de performanță.....	10
Modul de utilizare al programului.....	11
Ascultare melodie.....	11
Încărcare playlist.....	11
Ascultare post radio.....	11
Creare playlist.....	12
Modificare playlist.....	12
Ieșirea din aplicație.....	12
Diagramele UML.....	13
Diagramă de clase pentru modulul StateChange.....	13
Diagramă de clase pentru TestStateChange.....	14
Diagramă de clase pentru programul principal.....	15
Diagramă pentru cazurile de utilizare.....	16
Diagrama de secvențe atunci când utilizatorul deschide un fișier mp3.....	17
Diagrama de secvențe atunci când utilizatorul deschide un playlist.....	18
Diagrama de secvențe atunci când utilizatorul redă un post radio.....	19
Diagrama de secvențe atunci când utilizatorul crează un nou playlist.....	20
Diagrama de secvențe atunci când utilizatorul modifică un playlist.....	21
Diagrama de activități atunci când utilizatorul deschide un fișier mp3.....	22
Diagrama de activități atunci când utilizatorul deschide un playlist.....	23
Diagrama de activități atunci când utilizatorul redă un post radio.....	24
Diagrama de activități atunci când utilizatorul crează un playlist.....	25
Diagrama de activități atunci când utilizatorul modifică un playlist.....	26
Capturi de ecran din timpul folosirii aplicației.....	27
Redarea unei melodii individuale.....	27
Redarea unui playlist.....	28
Crearea unui playlist.....	29
Modificarea unui playlist.....	30

Ascultarea unui post radio.....	31
Afişarea fişierului help.....	31
Anexa 1.....	32
Cod reprezentativ pentru interfaţa IState.....	32
Cod reprezentativ pentru clasa Context.....	32
Cod reprezentativ pentru clasa EditPlaylistState.....	33
Cod reprezentativ pentru clasa MakePlaylistState.....	33
Cod semnificativ pentru enum-ul MP3PlayerStates.....	34
Cod semnificativ pentru clasa PlaylistState.....	34
Cod reprezentativ pentru clasa RadioState.....	35
Cod reprezentativ pentru clasa SingleFileState.....	35
Cod reprezentativ pentru clasa principală a programului.....	36

# 1. Scopul documentului

Scopul acestui document SRS (Software Requirements Specification) este de a defini cerințele funcționale și non-funcționale pentru dezvoltarea unui MP3 player. Acest document servește ca bază comună de înțelegere între echipele de dezvoltare, management și clienți cu privire la funcționalitățile și comportamentul sistemului MP3 player. Scopul final este de a asigura o dezvoltare coerentă și consistentă a produsului, îndeplinind așteptările utilizatorilor și cerințele de calitate ale sistemului MP3 player.

## 2. Prezentare generală a documentului

Restul documentului SRS este format din trei capitole, primul oferă o descriere generală a produsului software despre situația pe care dorește să o rezolve, scopul proiectului, contextul și beneficiile sale.

Al doilea capitol descrie cerințele funcționale pe care trebuie să le îndeplinească aplicația. Descrie utilizatorul ce folosește aplicația și cazurile de utilizare.

Al treilea capitol descrie cerințele ce nu se referă la funcționalitatea de bază, dar sunt importante pentru utilizare, cum ar fi interfața cu utilizatorul și performanța.

## 3. Descriere generală a produsului

### 3.1. Situația curentă

Produsul MP3 player are rolul de a oferi utilizatorilor o soluție convenabilă pentru redarea fișierelor MP3 individuale, gestionarea playlist-urilor și accesul la posturi de radio. În situația curentă, utilizatorii trebuie să utilizeze diferite dispozitive și aplicații separate pentru a îndeplini aceste funcționalități.

Prin utilizarea MP3 player-ului propus, utilizatorii vor putea avea toate aceste caracteristici integrate într-o singură aplicație ușor de utilizat. Aplicația va permite utilizatorilor să încarce și să redea fișierele MP3 preferate într-un mod simplu și intuitiv. Vor putea crea playlist-uri personalizate pentru a organiza și reda colecția lor de muzică într-un mod personalizat. De asemenea, vor avea acces la o varietate de posturi radio online, pe care le pot explora și reda direct pe aplicația TuneWave.

Prin eliminarea nevoii de a utiliza mai multe dispozitive și aplicații separate, TuneWave își propune să ofere utilizatorilor o experiență de utilizare simplificată și eficientă în domeniul redării fișierelor muzicale. Astfel, utilizatorii vor avea o soluție all-in-one pentru a se bucura de muzică într-un mod facil.

### **3.2. Scopul produsului**

Scopul TuneWave este de a oferi utilizatorilor o platformă integrată pentru redarea fișierelor MP3, gestionarea playlist-urilor, accesul la posturi de radio și crearea/modificarea playlist-urilor existente. Produsul urmărește să ofere o experiență de utilizare simplă, intuitivă și convenabilă pentru utilizatorii care doresc să se bucure de muzică în format digital. Scopul este de a satisface nevoile utilizatorilor în ceea ce privește redarea și organizarea muzicii, oferindu-le o soluție completă și eficientă pentru a-și gestiona colecția de fișiere audio și pentru a descoperi și reda posturi radio preferate.

### **3.3. Contextul produsului**

TuneWave este conceput pentru a se integra într-un mediu digital în continuă evoluție, în care redarea și gestionarea fișierelor audio sunt obiective importante pentru utilizatori. Produsul este destinat utilizatorilor de toate categoriile de vârstă și niveluri de expertiză tehnică, care doresc să se bucure de muzică în format digital.

### **3.4 Beneficii**

În ansamblu, TuneWave oferă utilizatorilor o experiență comodă și personalizată în redarea și gestionarea muzicii, aducând beneficii precum accesul simplificat la muzică, organizarea eficientă și posibilitatea de a crea o experiență de ascultare personalizată.

## 4. Cerinte funcționale

### 4.1. Actori

Utilizatorul este principalul actor și reprezintă persoana care utilizează aplicația TuneWave pentru a reda, gestiona și interacționa cu fișierele MP3, playlist-urile și posturile de radio. Utilizatorul poate crea playlist-uri, adăuga sau șterge fișiere MP3 din acestea, reda și opri redarea muzicii, controla volumul și alte setări audio etc.

### 4.2. Cazuri de utilizare

#### 4.2.1. Redarea unui fișier mp3 individual

Numele cazului de utilizare:	Redarea unui fișier mp3 individual
Rezumat:	Această operație permite unui utilizator să redea un fișier mp3 individual.
Prioritate:	Crucială
Declanșator:	Utilizatorul decide să redea un fișier mp3 individual.
Precondiție:	Utilizatorul trebuie să aibă instalat un sistem de operare ce oferă suport pentru obiectul COM AxWindowsMediaPlayer și să aibă un fișier mp3 pe care să-l redea.
Calea standard:	<ol style="list-style-type: none"><li>1. Utilizatorul deschide aplicația.</li><li>2. Utilizatorul apasă pe Meniu.</li><li>3. Utilizatorul apasă pe Deschidere fisier.</li><li>4. Utilizatorul selectă un fișier mp3 în dialog.</li><li>5. Se începe redarea fișierului selectat.</li></ol>
Parcurs alternativ:	Nu există
Post condiție:	Fișierul MP3 este redat cu succes, iar utilizatorul poate auzi muzica asociată fișierului selectat.
Calea cu excepții:	<ol style="list-style-type: none"><li>1. În cazul în care utilizatorul întrerupe manual redarea fișierului MP3 înainte de finalizare sau trece în altă stare, aplicația oprește redarea și utilizatorul revine la starea inițială, fiind pregătit pentru a reda alt fișier MP3 sau a efectua alte acțiuni în aplicație.</li><li>2. În cazul în care nu se selectează un fișier mp3 și se închide</li></ol>

	<p>dialogul, se revine în starea inițială.</p> <p>3. Dacă apar alte excepții în program, va fi afișat un MessageBox ce descrie excepția apărută.</p>
--	--

#### 4.2.2. Redarea unui playlist

Numele cazului de utilizare:	Redarea unui playlist
Rezumat:	Această operație permite unui utilizator să redea un playlist cu mai multe fișiere mp3.
Prioritate:	Crucială
Declanșator:	Utilizatorul decide să redea un playlist.
Precondiție:	Utilizatorul trebuie să aibă instalat un sistem de operare ce oferă suport pentru obiectul COM AxWindowsMediaPlayer și să aibă un playlist cu fișiere mp3 și fișierele la locația specificată în playlist.
Calea standard:	<ol style="list-style-type: none"> <li>1. Utilizatorul deschide aplicația.</li> <li>2. Utilizatorul apasă pe Meniu.</li> <li>3. Utilizatorul apasă pe Deschidere Playlist.</li> <li>4. Utilizatorul selectează un fișier valid pentru playlist.</li> <li>5. Se începe redarea playlist-ului.</li> </ol>
Parcurs alternativ:	Nu există
Post condiție:	Playlist-ul este redat, și utilizatorul poate asculta prima melodie. Dacă decide, poate să activeze opțiunea Loop, bifând o căsuță, și poate asculta în buclă aceeași melodie. Dacă decide, și a dezactivat opțiunea Loop, poate să redea melodiile într-o ordine alternativă, bifând căsuța cu numele Random. Dacă nu este bifată niciuna din căsuțe, redarea va avea loc secvențial, și se va reveni la prima melodie când se termină playlist-ul. De asemenea, utilizatorul poate alege melodia ce se va reda, selectând numele acesteia într-o listă.
Calea cu excepții:	<ol style="list-style-type: none"> <li>1. Dacă utilizatorul deschide un fișier txt ce nu corespunde unui playlist, i se va comunica că nu se cunoaște tipul de fișier. Pentru a ieși din această stare, utilizatorul trebuie să selecteze o altă opțiune din Meniu.</li> <li>2. Dacă apar alte excepții în program, va fi afișat un MessageBox ce descrie excepția apărută.</li> </ol>

### 4.2.3. Redarea unui post radio

Numele cazului de utilizare:	Redarea unui post radio
Rezumat:	Această operație permite unui utilizator să asculte un post radio din lista prestabilită.
Prioritate:	Crucială
Declanșator:	Utilizatorul decide să asculte un post radio.
Precondiție:	Utilizatorul trebuie să aibă instalat un sistem de operare ce oferă suport pentru obiectul COM AxWindowsMediaPlayer și să aibă o conexiune la internet pentru a reda posturile radio.
Calea standard:	<ol style="list-style-type: none"><li>1. Utilizatorul deschide aplicația.</li><li>2. Utilizatorul apasă pe Meniu.</li><li>3. Utilizatorul apasă pe Ascultare post radio.</li><li>4. Se redă primul post radio din lista predefinită.</li></ol>
Parcurgere alternativă:	Nu există
Post condiție:	Utilizatorul poate asculta primul post radio. Acesta poate alege unul din posturile radio predefinite, selectând numele acestuia într-o listă.
Calea cu excepții:	<ol style="list-style-type: none"><li>1. În cazul în care utilizatorul nu are acces la internet, nu se va putea reda conținutul postului radio.</li><li>2. Dacă apar alte excepții în program, va fi afișat un MessageBox ce descrie excepția apărută.</li></ol>

### 4.2.4. Crearea unui playlist

Numele cazului de utilizare:	Crearea unui playlist
Rezumat:	Această operație permite unui utilizator crearea unui nou playlist format din unul sau mai multe fișiere mp3.
Prioritate:	Crucială
Declanșator:	Utilizatorul decide să creeze un nou playlist.
Precondiție:	Utilizatorul trebuie să aibă instalat un sistem de operare ce oferă suport pentru obiectul COM AxWindowsMediaPlayer și să aibă fișiere mp3 pe care le poate adăuga la noul playlist.
Calea standard:	<ol style="list-style-type: none"><li>1. Utilizatorul deschide aplicația.</li><li>2. Utilizatorul apasă pe Meniu.</li></ol>



	<p>3. Utilizatorul apasă pe Creare playlist.</p> <p>4. Utilizatorul adaugă melodii apăsând butonul Adaugă și acestea vor apărea mai jos.</p> <p>5. După ce a adăugat toate melodiile dorite, utilizatorul poate apăsa butonul Salvează pentru a salva noul playlist.</p> <p>6. Utilizatorul scrie în caseta de dialog numele fișierului rezultat sau suprascrive alt fișier.</p> <p>7. Se elimină controalele, sistemul revenind în starea inițială.</p> <p>7. Noul playlist este disponibil în fișierul creat de utilizator.</p>
Parcurgere alternativă:	Folosind funcția de modificare a unui playlist și salvând fișierul cu alt nume.
Post condiție:	Fișierul scris de utilizator poate fi redat folosind funcția de redare playlist.
Calea cu excepții:	<p>1. Dacă utilizatorul nu salvează fișierul, atunci se elimină ce era completat în această stare.</p> <p>2. Dacă utilizatorul nu selectează niciun fișier și închide căsuța de dialog, nu se va întâmpla nimic.</p> <p>3. Dacă apar alte excepții în program, va fi afișat un MessageBox ce descrie excepția apărută.</p>

#### 4.2.5. Modificarea unui playlist existent

Numele cazului de utilizare:	Modificarea unui playlist existent
Rezumat:	Utilizatorul poate să modifice un playlist deja creat, adăugând sau ștergând melodii din acesta.
Prioritate:	Crucială
Declanșator:	Utilizatorul dorește să modifice un playlist existent.
Precondiție:	Utilizatorul trebuie să aibă instalat un sistem de operare ce oferă suport pentru obiectul COM AxWindowsMediaPlayer și să aibă un playlist pe care să-l modifice.
Calea standard:	<p>1. Utilizatorul deschide aplicația.</p> <p>2. Utilizatorul apasă pe Meniu.</p> <p>3. Utilizatorul apasă pe Modificare playlist existent.</p> <p>4. Utilizatorul selectează un fișier playlist.</p> <p>5. În lista de elemente se afișează melodiile disponibile în playlist.</p> <p>6. Utilizatorul poate adăuga melodii apăsând butonul Adăugare fișier, după care trebuie să selecteze un fișier mp3.</p> <p>7. Utilizatorul poate să șteargă elementul selectat din listă apăsând butonul Ștergere index selectat.</p> <p>8. Utilizatorul poate salva fișierul modificat apăsând pe butonul</p>

	Salvează playlist modificat. 9. După ce s-a salvat fișierul, se elimină controalele de pe interfață, și se pot efectua alte operații în program.
Parcurgere alternativă:	Folosind funcția de creere a unui playlist și salvând fișierul, suprascriind un fișier existent.
Post condiție:	Playlistul salvat de utilizator poate fi redat cu funcția de redare playlist.
Calea cu excepții:	<ol style="list-style-type: none"> <li>1. Dacă utilizatorul renunță la adăugarea unui fișier, nu se va întâmpla nimic.</li> <li>2. Dacă utilizatorul salvează un playlist fără niciun element, la redare se va afișa un MessageBox care să explice imposibilitatea redării unui playlist fără elemente.</li> <li>3. Dacă utilizatorul nu salvează ce era în playlist-ul nou creat, nu se va întâmpla nimic.</li> <li>4. Dacă apar alte excepții în program, va fi afișat un MessageBox ce descrie excepția apărută.</li> </ol>

## 5. Cerinte nefuncționale

### 5.1. Cerințe pentru interfața cu utilizatorul

Interfața cu utilizatorul trebuie să fie intuitivă, atractivă și să ofere o navigare și organizare eficientă a funcționalităților aplicației MP3 player.

### 5.2. Cerințe de performanță

Sistemul ar trebui să funcționeze în timp real: orice operație de redare sau modificare de playlist-uri, produsă de utilizator, ar trebui să se termine în mai puțin de 10 secunde.

Sistemul ar trebui să permită playlist-uri cu măcar 20 de melodii.

# Modul de utilizare al programului

## Ascultare melodie

- Pentru a asculta o melodie, apasă pe Meniu, apoi pe Deschidere fișier și selectează un fișier mp3.
- În acest mod, melodia selectată este redată în buclă.

## Încărcare playlist

- Pentru a reda un playlist, accesează meniul principal și selectează opțiunea "Deschidere playlist".
- Alege un fișier de tip playlist și vei putea reda melodiile conținute în acesta.
- Melodiile sunt redare în mod secvențial, respectând ordinea din playlist. După finalizarea playlist-ului, redarea se reia de la început.
- În timpul redării, poți selecta o melodie specifică apăsând pe numele acesteia.
- Poți activa funcția de redare în buclă (loop) pentru a reda una din melodiile din playlist în mod repetat. Această funcție poate fi activată prin selectarea casetei "Loop".
- De asemenea, ai opțiunea de a reda melodiile într-o ordine aleatoare, dacă nu este activată funcția de mai sus de redare în buclă. Pentru a utiliza această opțiune, selectează caseta "Random", iar caseta "Loop" trebuie să fie debifată.

## Ascultare post radio

- Pentru a asculta un post radio, accesează meniul principal și selectează opțiunea "Ascultare post radio".
- Se va reda automat primul post din lista predefinită de posturi radio.
- Pentru a selecta un alt post radio, apasă pe numele acestuia în listă.
- Asigură-te că ai conexiune la internet activată pentru a utiliza această opțiune, deoarece posturile de radio sunt preluate de pe internet.
- În funcție de factorii tehnici, redarea unui post radio poate începe după o durată de până la 10 secunde.

## Creare playlist

- Pentru a crea un playlist, accesează meniul principal și selectează opțiunea "Creare playlist".
- Pentru a adăuga melodii, utilizează butonul "Adaugă" și selectează melodia pe care dorești să o incluzi în playlist.
- După ce ai terminat de selectat melodiile dorite, apasă pe butonul "Salvează" și introdu un nume pentru noul playlist.
- După salvare, interfața se va actualiza, permițându-ți să selectezi o nouă operație din meniu pentru a continua utilizarea aplicației.

## Modificare playlist

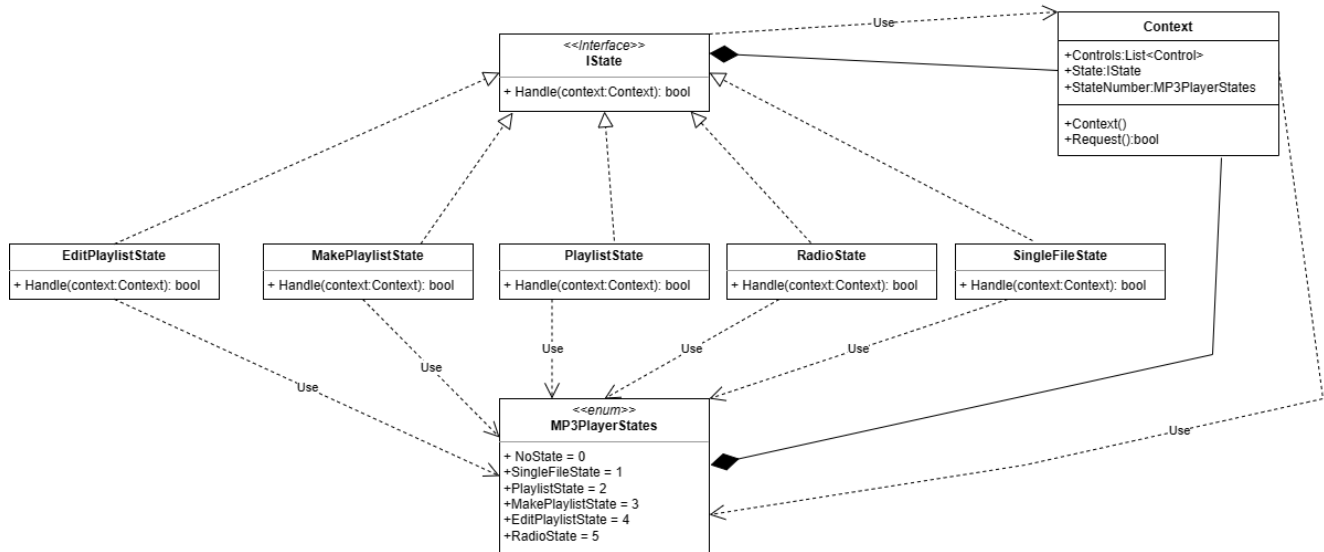
- Pentru a modifica un playlist existent, accesează meniul principal și selectează opțiunea "Modificare playlist existent".
- Ți se va solicita să selectezi fișierul care conține playlist-ul pe care dorești să-l modifice.
- După selectare, vei vedea conținutul playlist-ului într-o listă.
- Pentru a adăuga o nouă melodie, utilizează butonul "Adăugare fișier".
- Pentru a șterge o melodie din playlist, selectează numele melodiei și apoi apasă butonul "Ștergere index selectat".
- Pentru a salva playlist-ul modificat, apasă butonul "Salvează playlist modificat" și introduce un nume nou pentru playlist sau selectează fișierul inițial pe care l-ai modificat.
- După salvare, elementele vor fi eliminate din interfață și vei putea efectua alte operațiuni în aplicație.

## Ieșirea din aplicație

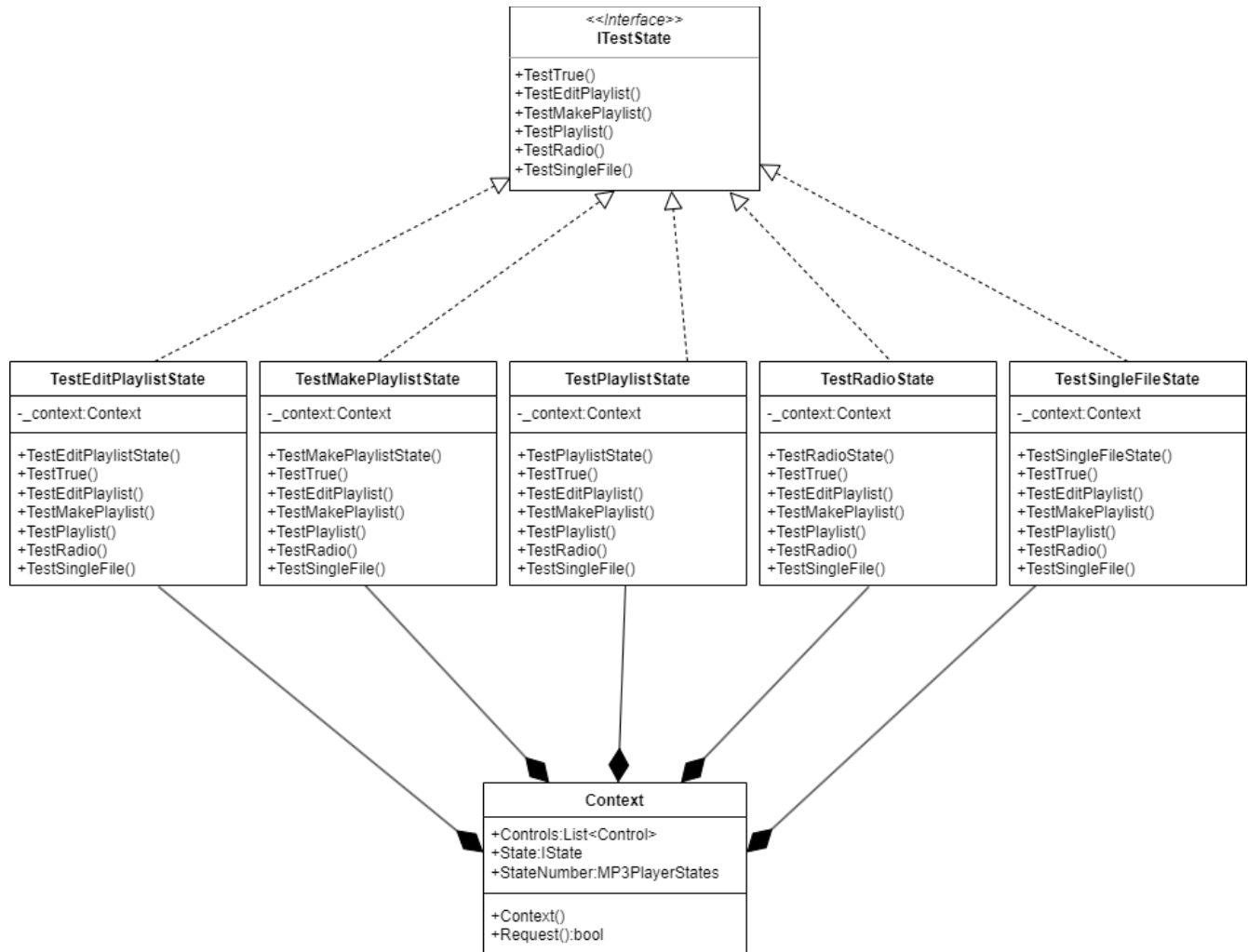
- Pentru a ieși din aplicație, selectează opțiunea "Ieșire" din meniu sau apasă pe butonul X al ferestrei.

# Diagramele UML

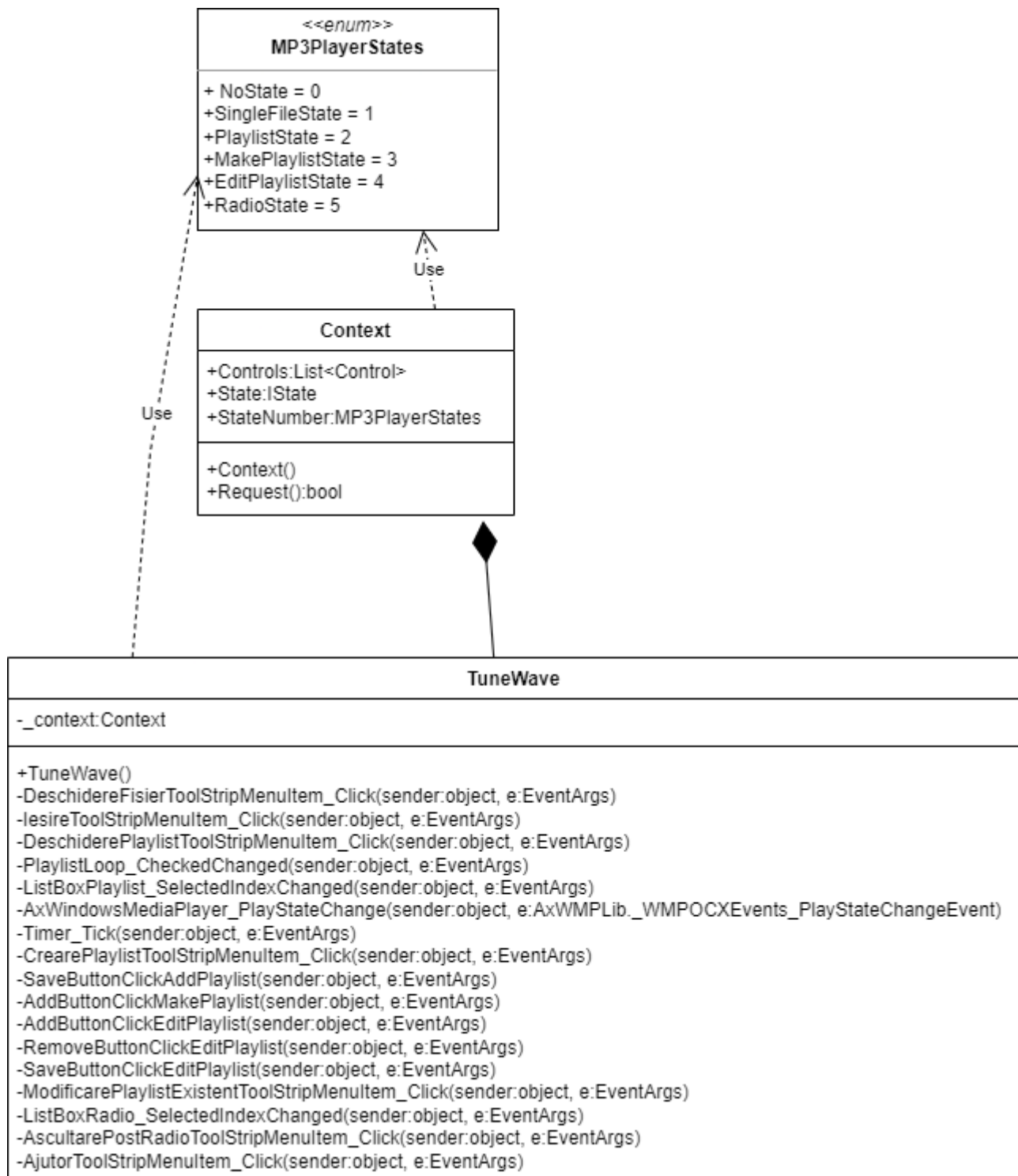
## Diagramă de clase pentru modulul StateChange



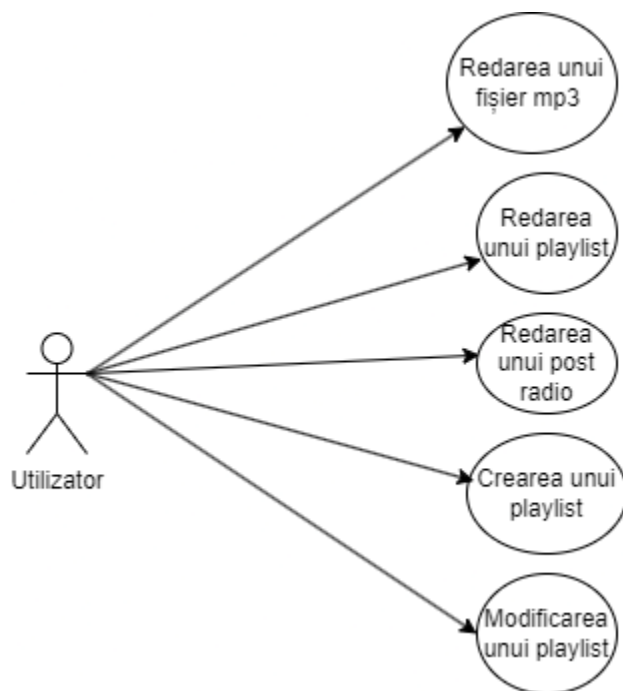
## Diagramă de clase pentru TestStateChange



## Diagramă de clase pentru programul principal

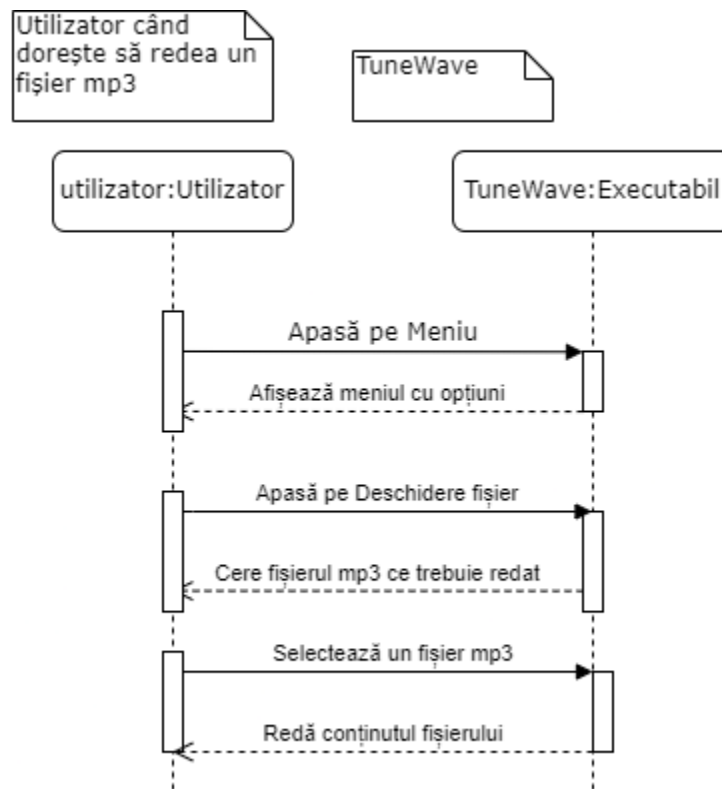


## Diagramă pentru cazurile de utilizare

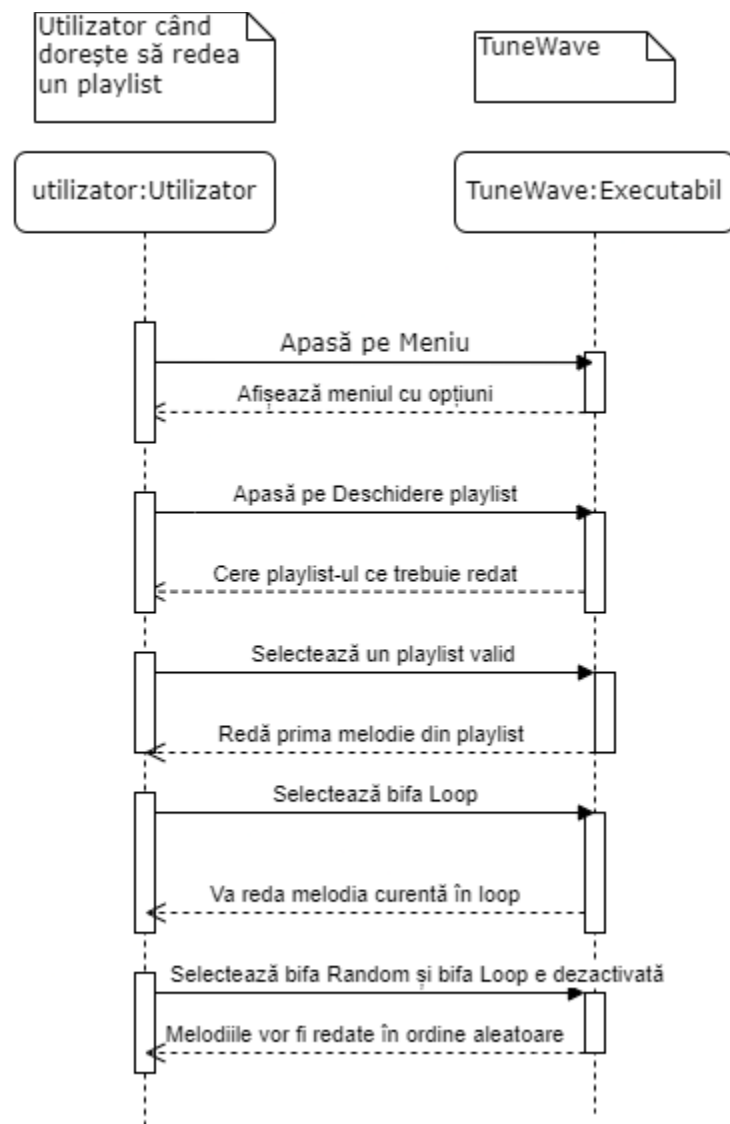




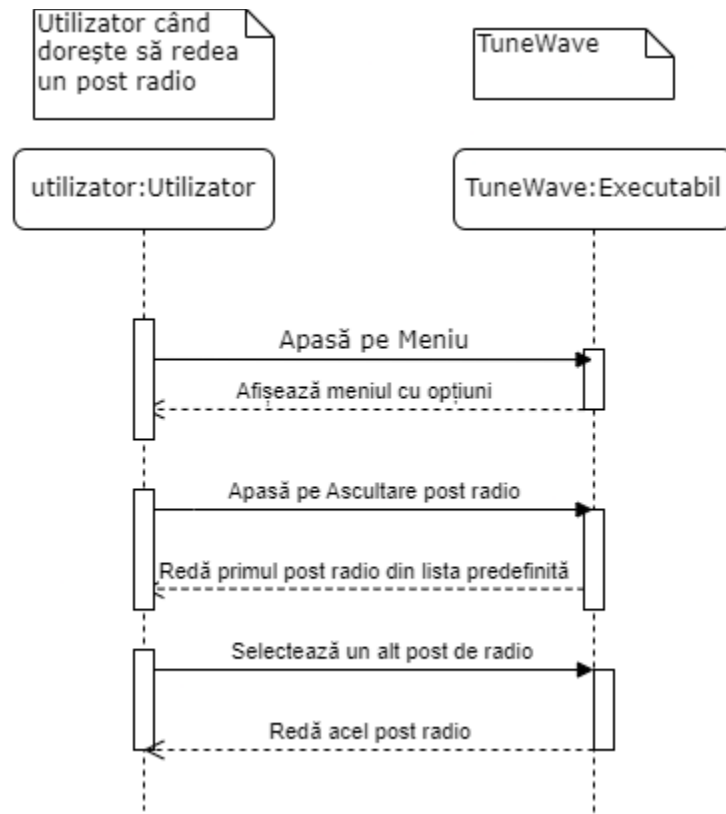
## Diagrama de secvențe atunci când utilizatorul deschide un fișier mp3



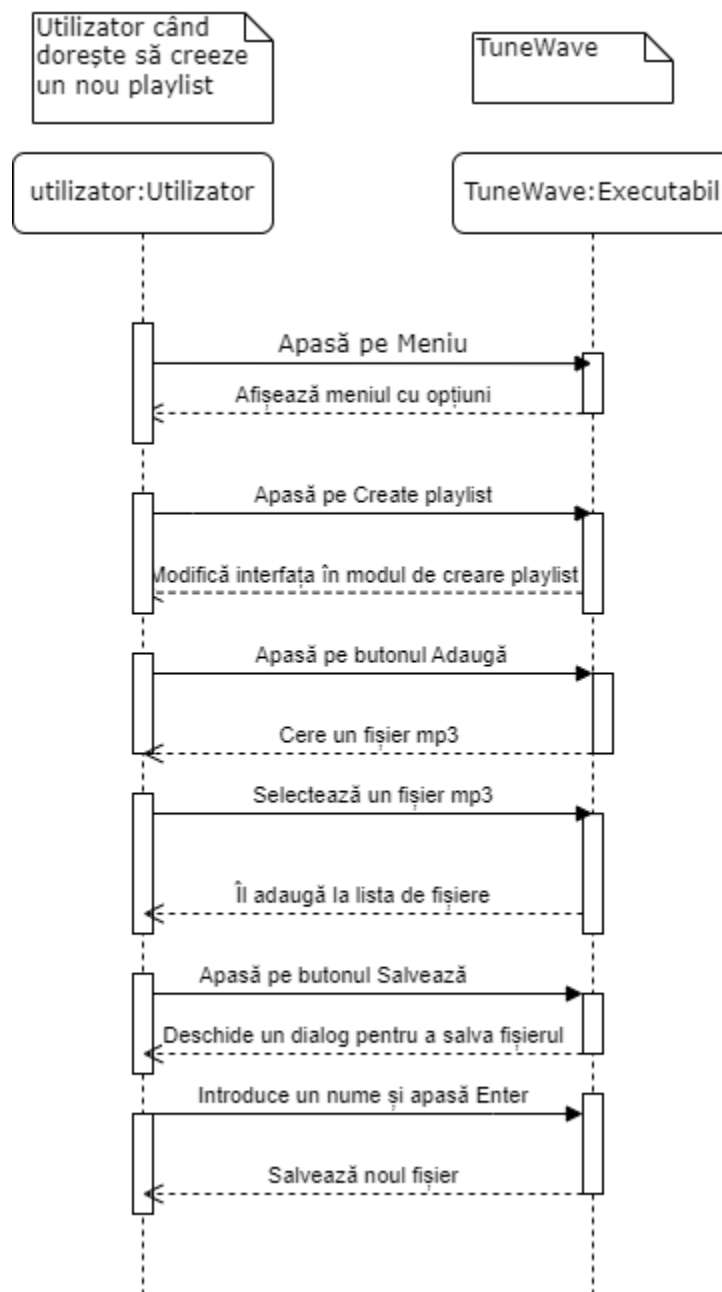
## Diagrama de secvențe atunci când utilizatorul deschide un playlist



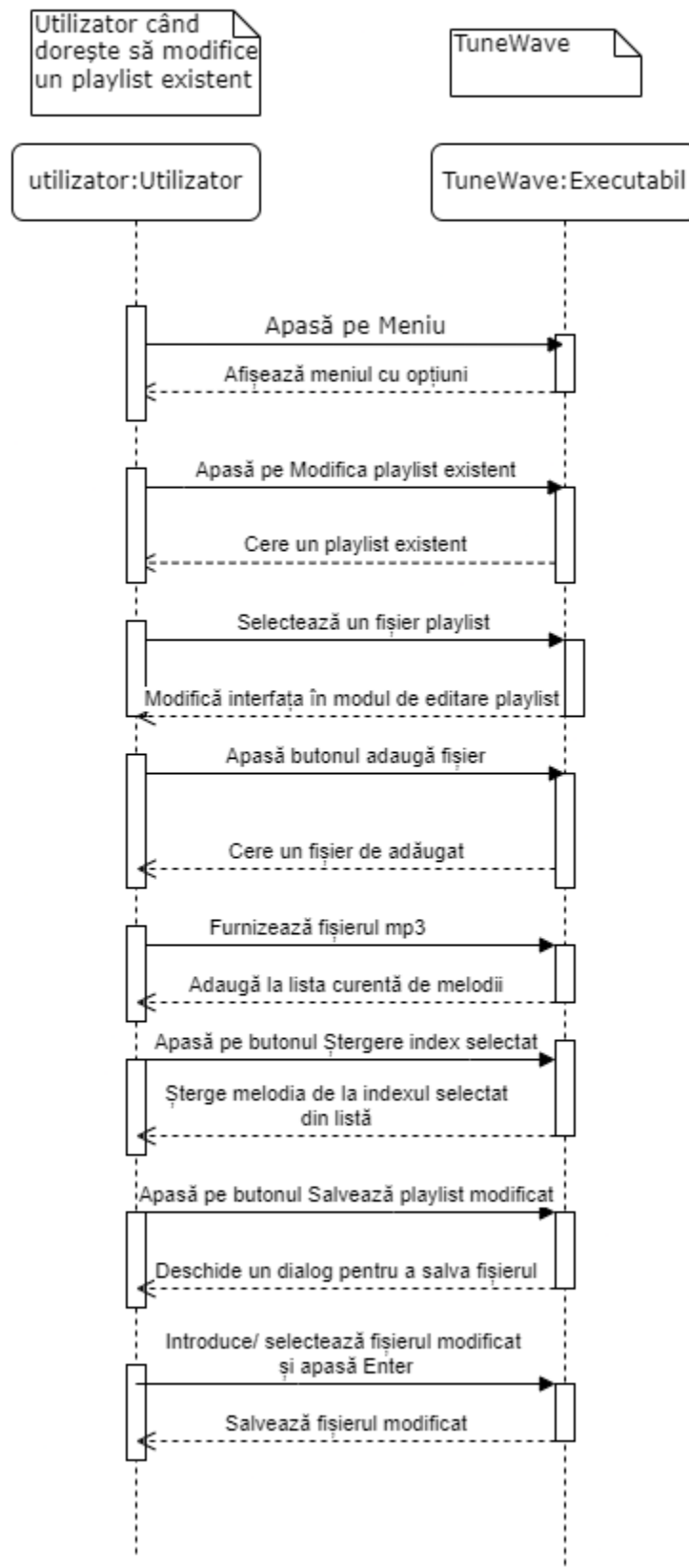
## Diagrama de secvențe atunci când utilizatorul redă un post radio



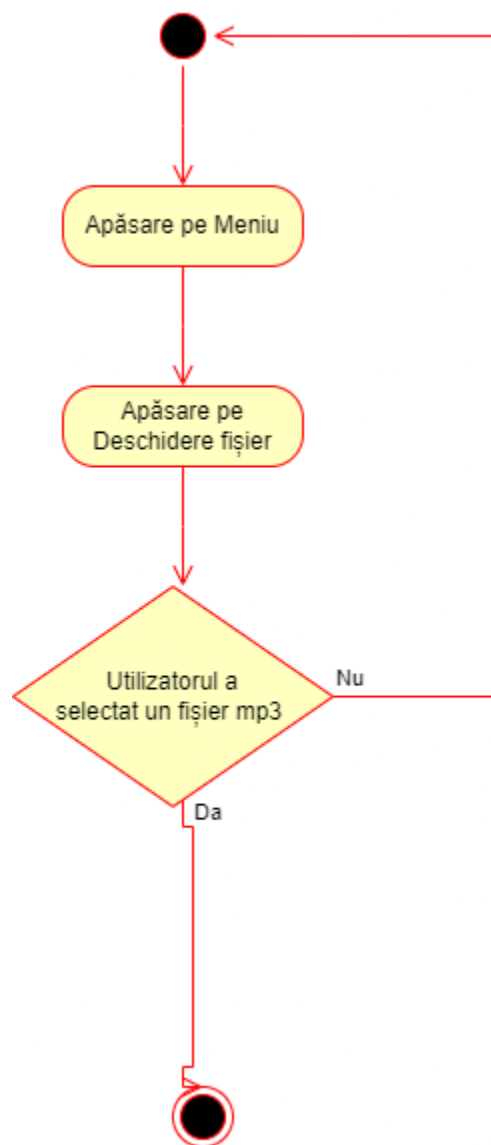
## Diagrama de secvențe atunci când utilizatorul crează un nou playlist



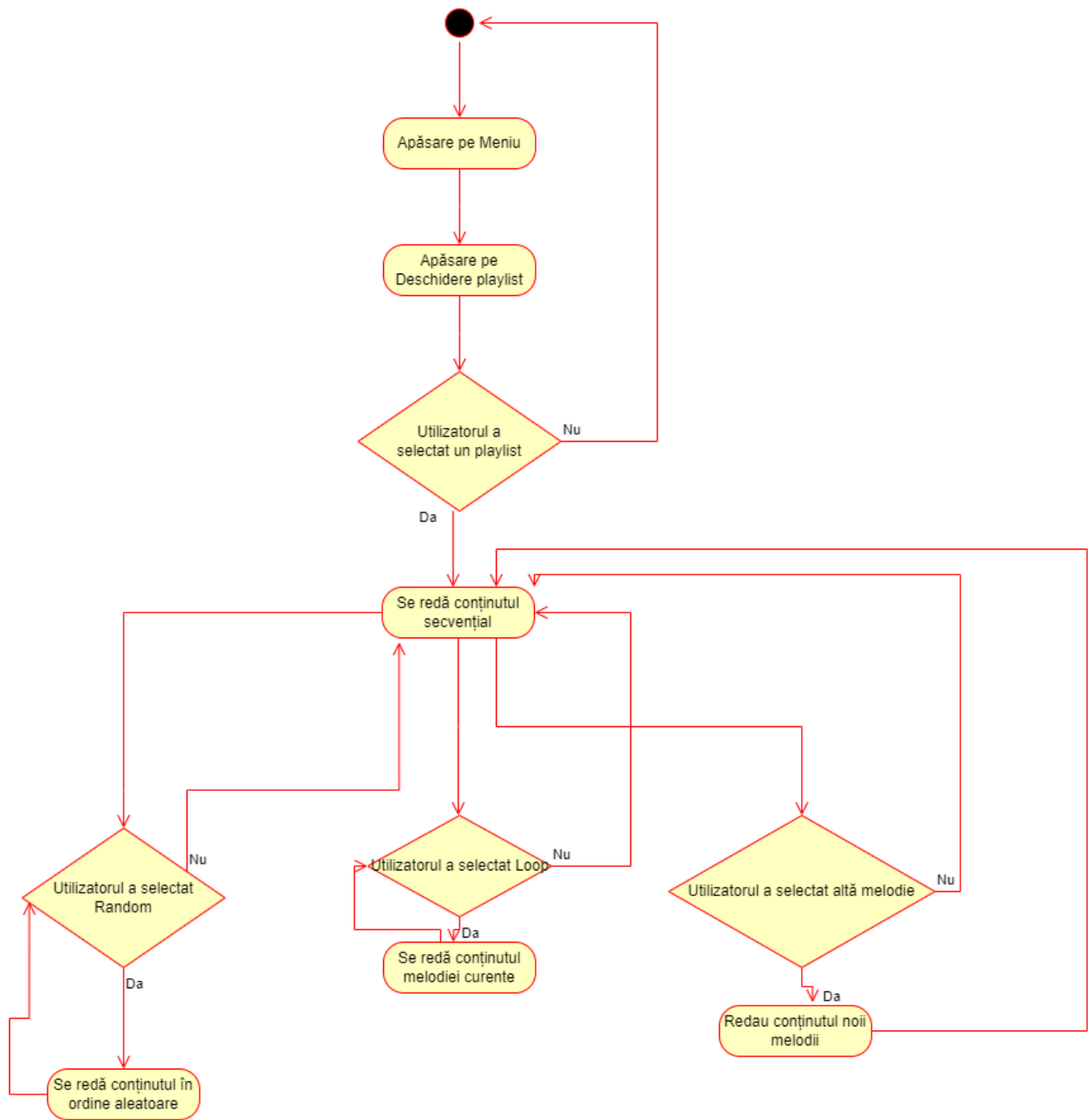
## Diagrama de secvențe atunci când utilizatorul modifică un playlist



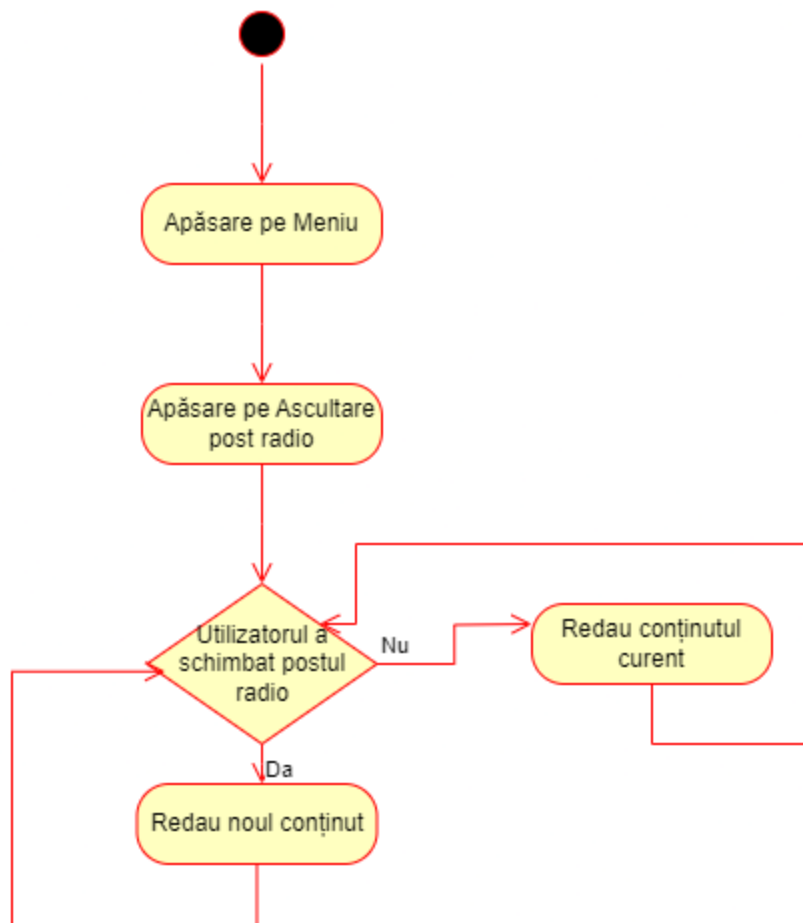
## Diagrama de activități atunci când utilizatorul deschide un fișier mp3



## Diagrama de activități atunci când utilizatorul deschide un playlist

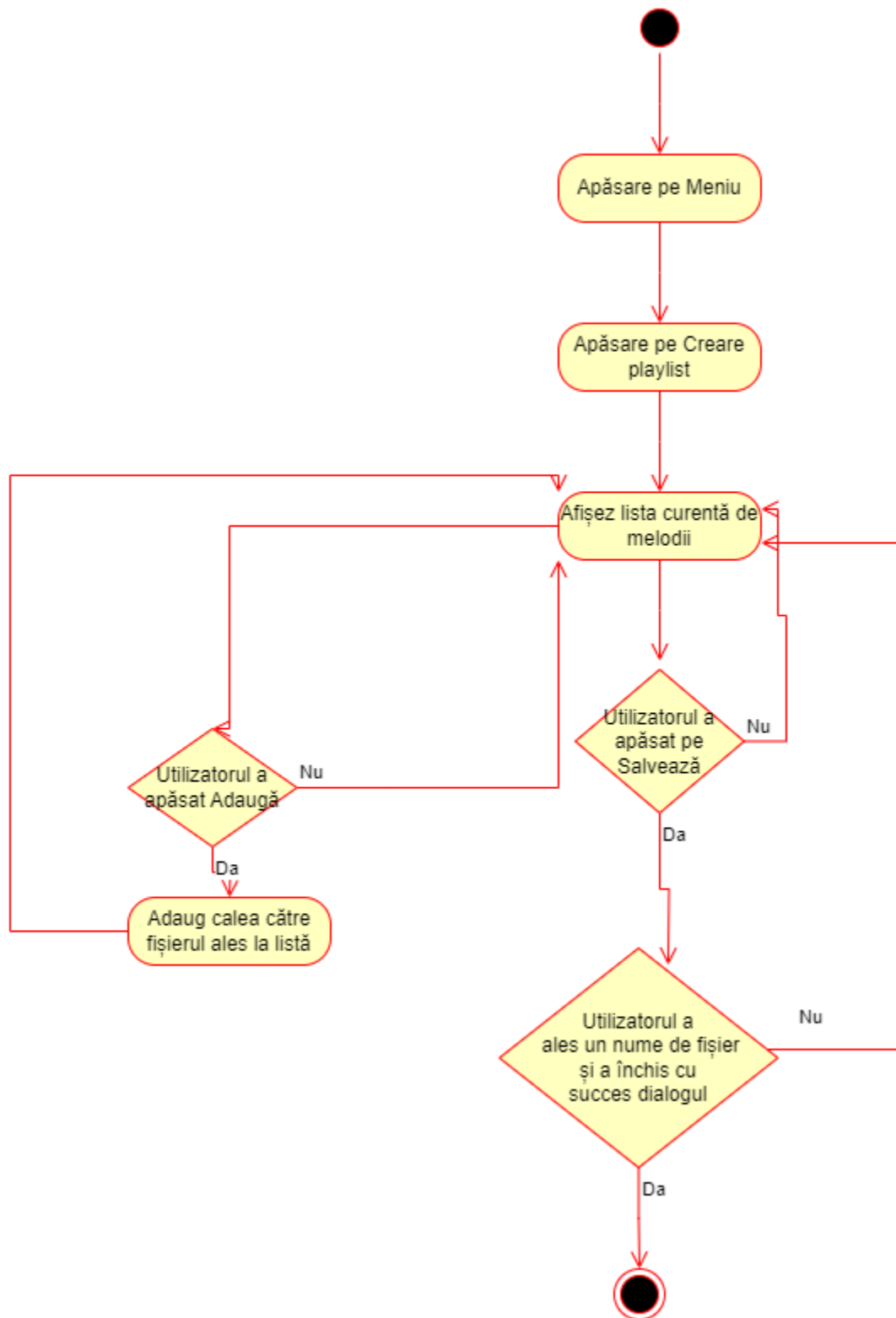


## Diagrama de activități atunci când utilizatorul redă un post radio

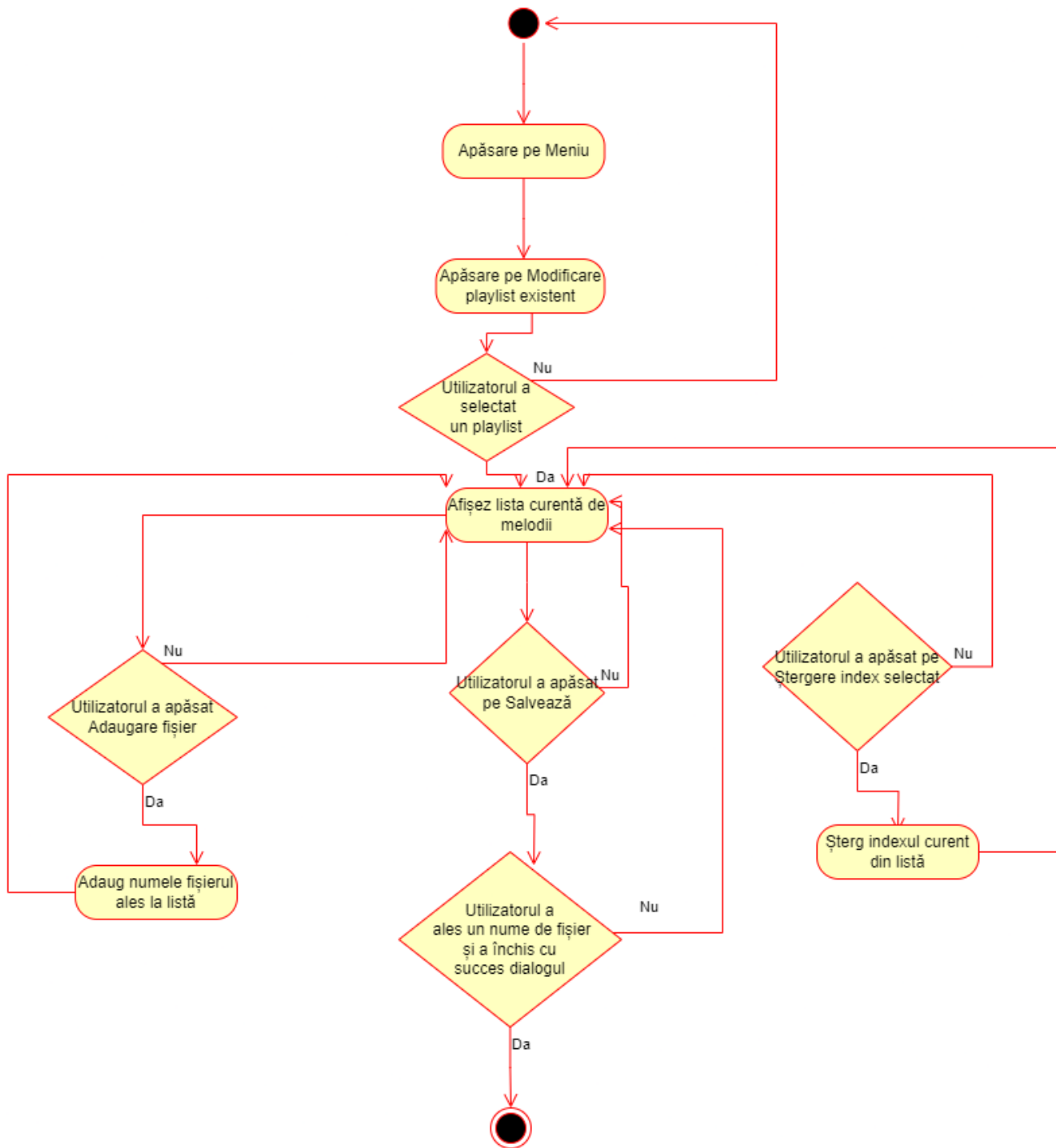




## Diagrama de activități atunci când utilizatorul crează un playlist

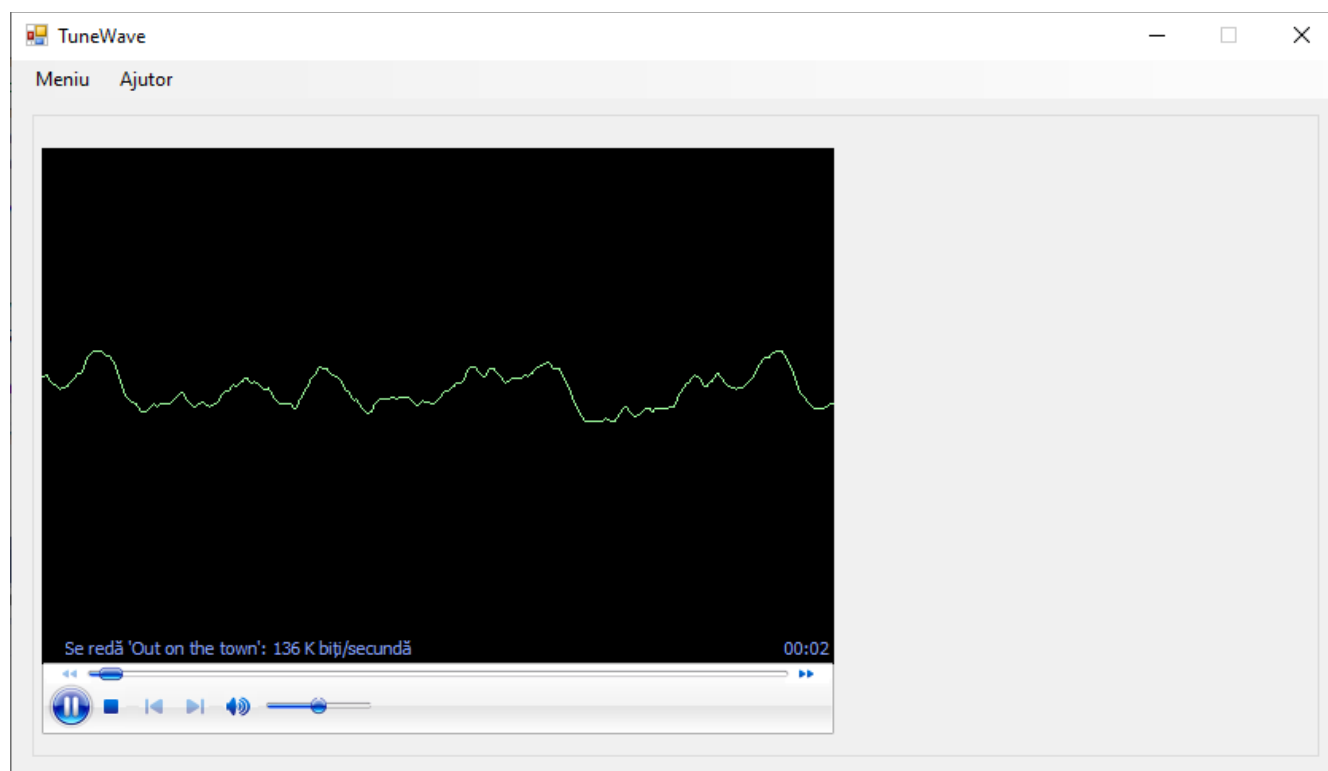


## Diagrama de activități atunci când utilizatorul modifică un playlist

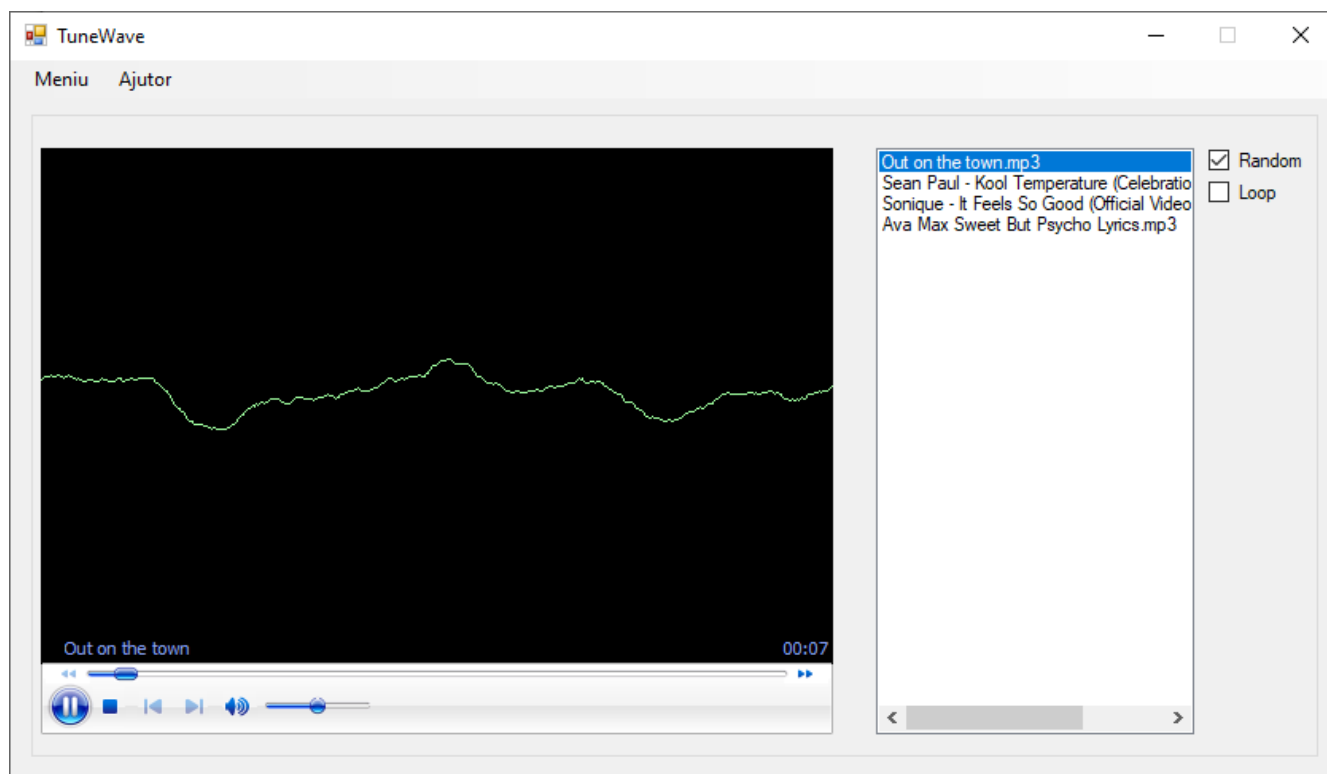


# Capturi de ecran din timpul folosirii aplicației

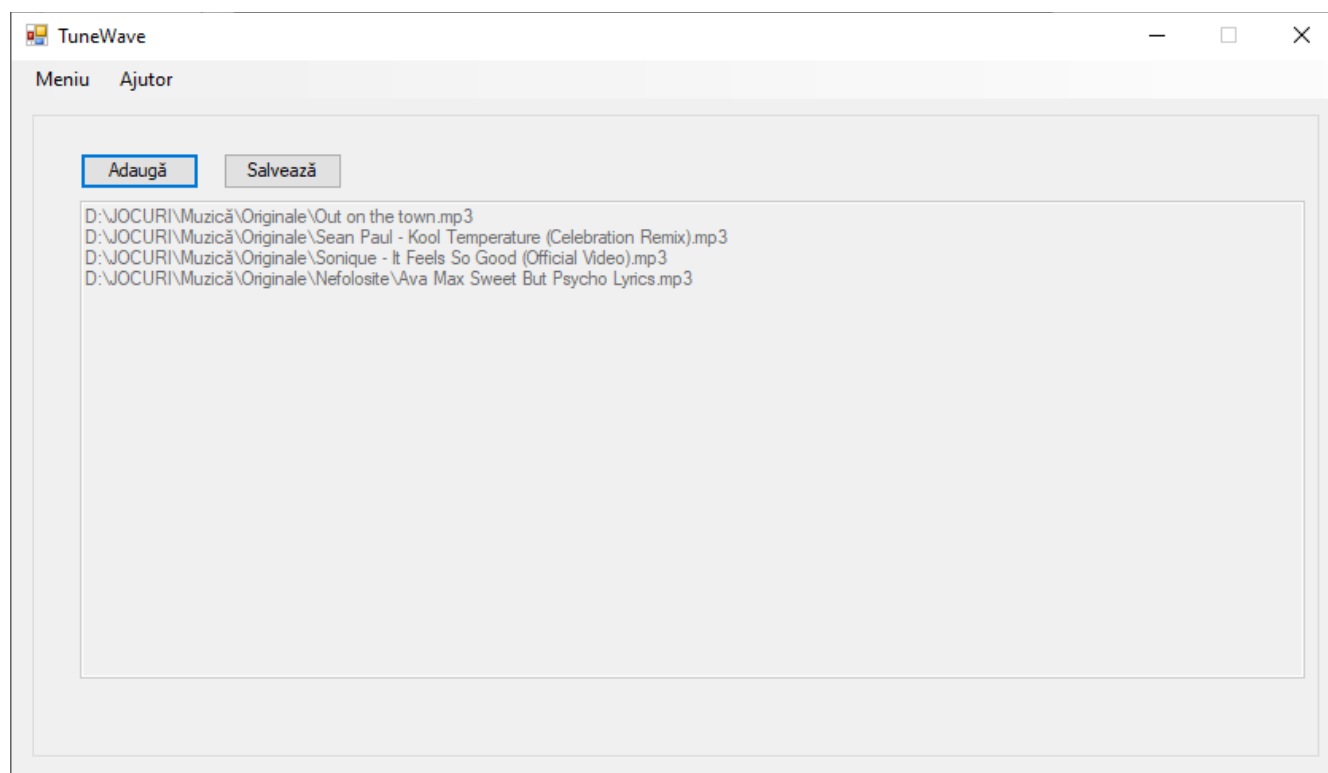
## Redarea unei melodii individuale



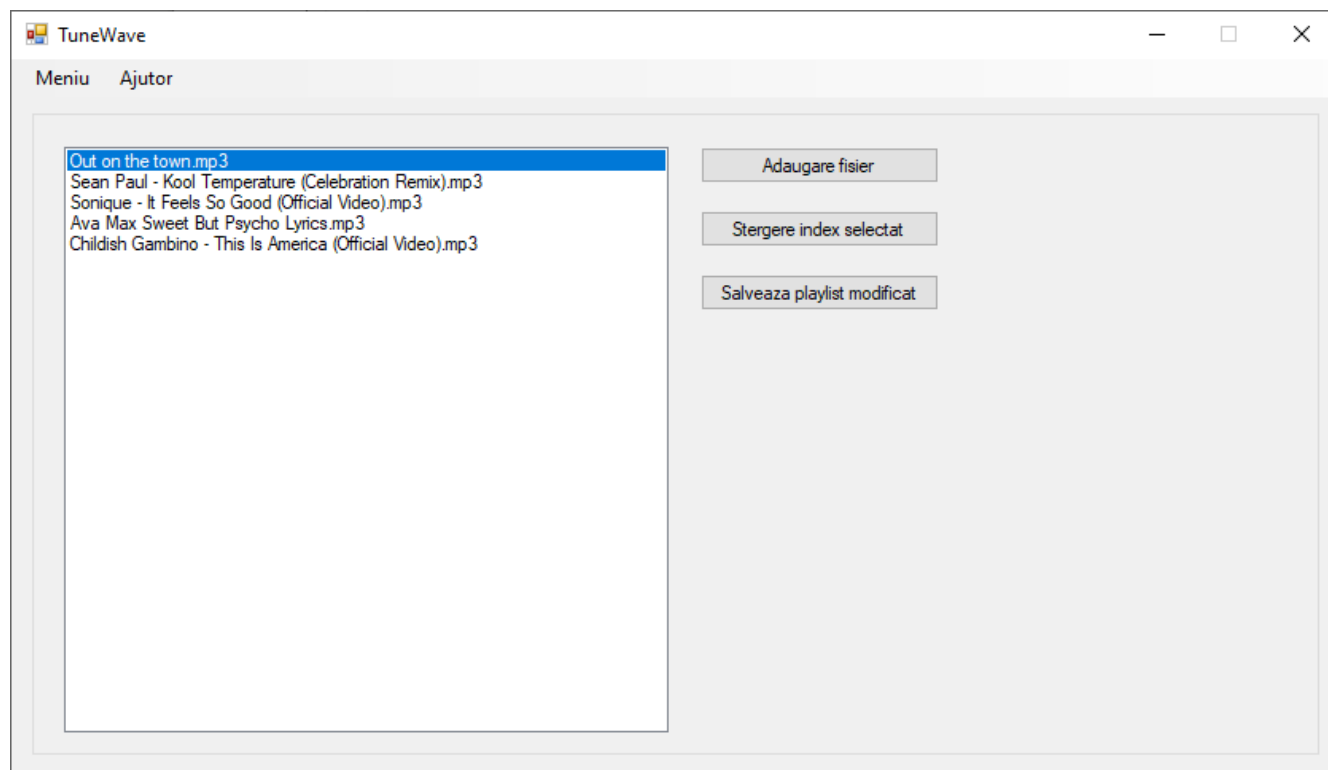
## Redarea unui playlist



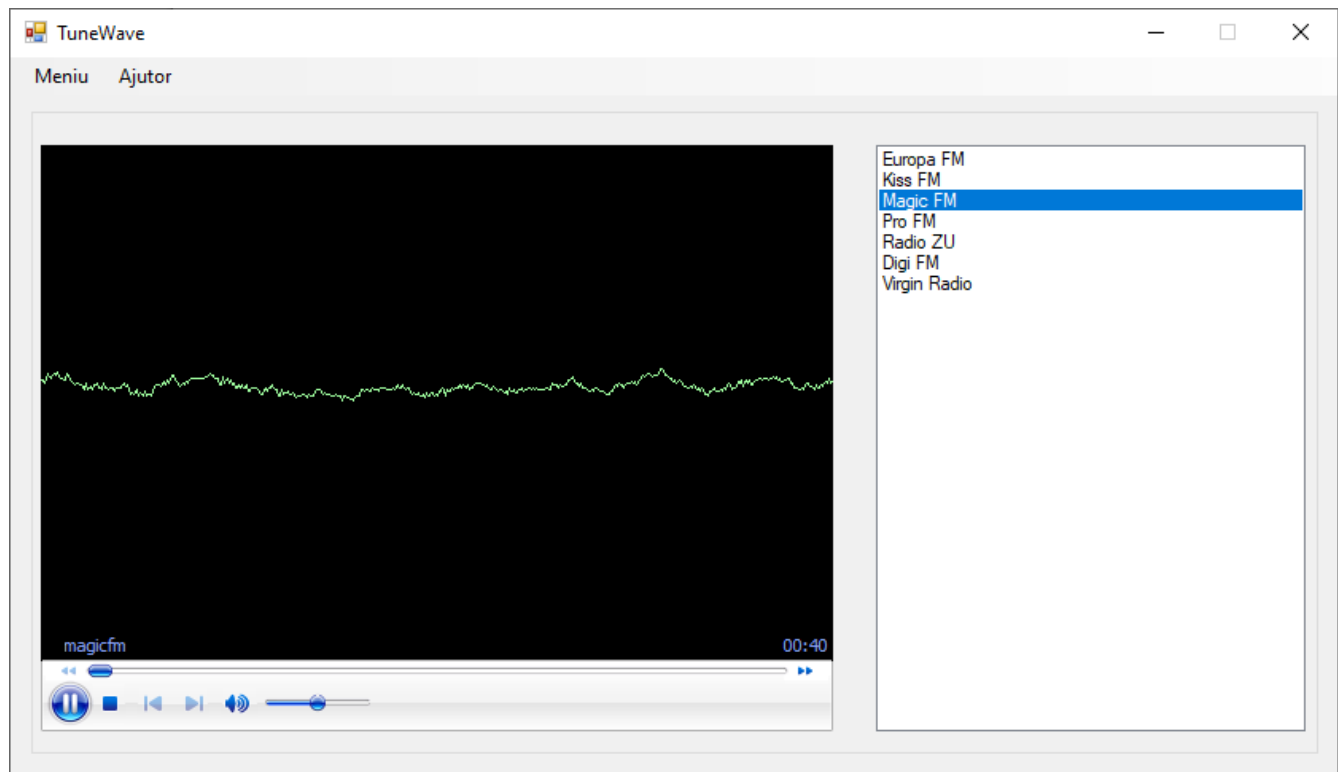
## Crearea unui playlist



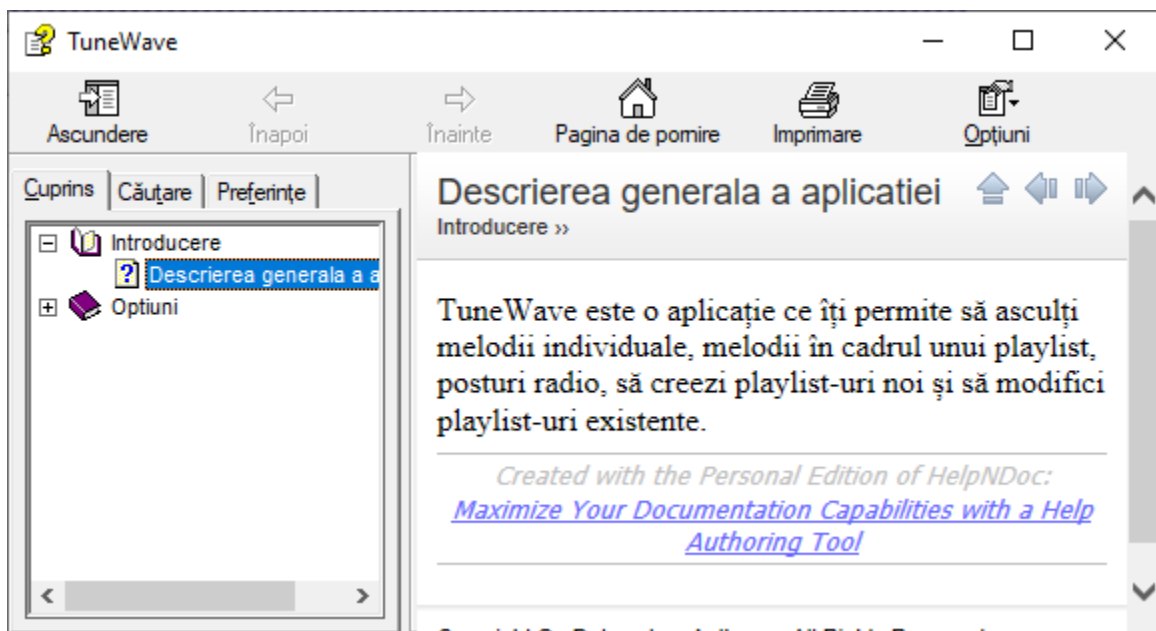
## Modificarea unui playlist



## Ascultarea unui post radio



## Afișarea fișierului help



# Anexa 1

## Cod reprezentativ pentru interfața IState

```
// Interfata folosita de context pentru a-și schimba comportamentul
public interface IState
{
    // Functie implementata in clasele derivate
    bool Handle(Context context);
}
```

## Cod reprezentativ pentru clasa Context

```
public class Context
{
    // Lista de controale pentru starea curenta
    public List<Control> Controls { get; set; }
    // Starea curenta a contextului
    public IState State { get; set; }
    // Numarul starii curente
    public MP3PlayerStates StateNumber { get; set; }
    // Constructorul Context initializeaza starea curenta, lista de controale si
    numarul starii cu "NoState"
    public Context()
    {
        State = new SingleFileState();
        Controls = new List<Control>();
        StateNumber = MP3PlayerStates.NoState;
    }
    // Metoda Request este folosita pentru a solicita incarcarea controalelor
    public bool Request()
    {
        //Se incearca incarcarea controalelor cu starea curenta
        if (!State.Handle(this))
        {
            //Daca s-a revenit pe false, inseamna ca s-a schimbat starea si trebuie
            facuta incarcarea controalelor
            State.Handle(this);
        }
        return true;
    }
}
```



## Cod reprezentativ pentru clasa EditPlaylistState

```
public bool Handle(Context context)
{
    // Se verifică dacă starea curentă este cea dorită pentru a fi afișată
    if (context.StateNumber == MP3PlayerStates.EditPlaylistState)
    {
        // Dacă da, se elimină controalele existente și se adaugă controalele
        pentru starea curentă
        context.Controls.Clear();
        context.Controls.Add(new Button());
        context.Controls.Add(new Button());
        context.Controls.Add(new ListBox());
        context.Controls.Add(new Button());
        return true;
    }
    // Dacă nu, se schimbă starea curentă într-o altă stare, în funcție de starea
    curentă
    switch (context.StateNumber)
    {
        case MP3PlayerStates.PlaylistState: context.State = new PlaylistState();
        break;
        case MP3PlayerStates.SingleFileState: context.State = new
        SingleFileState(); break;
        case MP3PlayerStates.MakePlaylistState: context.State = new
        MakePlaylistState(); break;
        case MP3PlayerStates.RadioState: context.State = new RadioState(); break;
    }
    return false;
}
```

## Cod reprezentativ pentru clasa MakePlaylistState

```
public bool Handle(Context context)
{
    // Verificăm dacă starea actuală este deja MakePlaylistState
    if (context.StateNumber == MP3PlayerStates.MakePlaylistState)
    {
        // Dacă da, eliminăm controalele existente și adăugăm cele necesare pentru
        această stare
        context.Controls.Clear();
        context.Controls.Add(new Button());
        context.Controls.Add(new Button());
        context.Controls.Add(new TextBox());
        return true;
    }

    // Dacă nu suntem în starea MakePlaylistState, trecem la altă stare
    switch (context.StateNumber)
```

```

    {
        case MP3PlayerStates.SingleFileState:
            context.State = new SingleFileState();
            break;
        case MP3PlayerStates.PlaylistState:
            context.State = new PlaylistState();
            break;
        case MP3PlayerStates.EditPlaylistState:
            context.State = new EditPlaylistState();
            break;
        case MP3PlayerStates.RadioState:
            context.State = new RadioState();
            break;
    }

    return false;
}
}

```

## Cod semnificativ pentru enum-ul MP3PlayerStates

```

public enum MP3PlayerStates
{
    // Stare inexistentă, folosită înainte de a se face initializarea
    NoState = 0,
    // Stare pentru redarea unui singur fisier audio
    SingleFileState = 1,
    // Stare pentru redarea unei liste de fisiere audio
    PlaylistState = 2,
    // Stare pentru crearea unei liste de fisiere audio
    MakePlaylistState = 3,
    // Stare pentru editarea unei liste de fisiere audio
    EditPlaylistState = 4,
    // Stare pentru redarea unui post de radio
    RadioState = 5
}

```

## Cod semnificativ pentru clasa PlaylistState

```

public bool Handle(Context context)
{
    // Verificăm dacă starea actuală este deja PlaylistState
    if (context.StateNumber == MP3PlayerStates.PlaylistState)
    {
        // Dacă da, eliminăm controalele existente și adăugăm cele necesare pentru
        această stare
        context.Controls.Clear();
        context.Controls.Add(new AxWindowsMediaPlayer());
    }
}

```

```

        context.Controls.Add(new ListBox());
        context.Controls.Add(new CheckBox());
        context.Controls.Add(new CheckBox());
        return true;
    }
    // Dacă nu suntem în starea PlaylistState, trecem la altă stare
    switch (context.StateNumber)
    {
        case MP3PlayerStates.SingleFileState: context.State = new
SingleFileState(); break;
        case MP3PlayerStates.MakePlaylistState: context.State = new
MakePlaylistState(); break;
        case MP3PlayerStates.EditPlaylistState: context.State = new
EditPlaylistState(); break;
        case MP3PlayerStates.RadioState: context.State = new RadioState(); break;
    }
    return false;
}

```

## Cod reprezentativ pentru clasa RadioState

```

public bool Handle(Context context)
{
    // Verificăm dacă starea curentă este "RadioState"
    if (context.StateNumber == MP3PlayerStates.RadioState)
    {
        // Curățăm controlerele și adăugăm noi controale pentru redarea radio-ului
și afișarea listei de stații
        context.Controls.Clear();
        context.Controls.Add(new AxWindowsMediaPlayer());
        context.Controls.Add(new ListBox());
        return true;
    }

    // Dacă starea curentă nu este "RadioState", trecem la o altă stare în funcție
de numărul stării curente
    switch (context.StateNumber)
    {
        case MP3PlayerStates.SingleFileState: context.State = new
SingleFileState(); break;
        case MP3PlayerStates.MakePlaylistState: context.State = new
MakePlaylistState(); break;
        case MP3PlayerStates.EditPlaylistState: context.State = new
EditPlaylistState(); break;
        case MP3PlayerStates.PlaylistState: context.State = new PlaylistState();
break;
    }
    return false;
}

```

## Cod reprezentativ pentru clasa SingleFileState

```
public bool Handle(Context context)
{
    // Verificăm dacă starea actuală este deja SingleFileState
    if (context.StateNumber == MP3PlayerStates.SingleFileState)
    {
        // Dacă da, eliminăm controalele existente și adăugăm cele necesare pentru
această stare
        context.Controls.Clear();
        AxWindowsMediaPlayer axWindowsMediaPlayer = new AxWindowsMediaPlayer();
        context.Controls.Add(axWindowsMediaPlayer);
        return true;
    }
    // Dacă nu suntem în starea SingleFileState, trecem la altă stare
    switch (context.StateNumber)
    {
        case MP3PlayerStates.PlaylistState: context.State = new PlaylistState();
break;
        case MP3PlayerStates.MakePlaylistState: context.State = new
MakePlaylistState(); break;
        case MP3PlayerStates.EditPlaylistState: context.State = new
EditPlaylistState(); break;
        case MP3PlayerStates.RadioState: context.State = new RadioState(); break;
    }
    return false;
}
```

## Cod reprezentativ pentru clasa principală a programului

```
//Funcție apelată atunci când se apasă "Deschidere fișier"
private void DeschidereFișierToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        //Deschid un dialog și citesc fișierul dat de la utilizator. Dacă nu am
fișier valid, parasesc funcția
        openFileDialog.Filter = "Audio file(*.mp3)|*.mp3";
        if (openFileDialog.ShowDialog() != DialogResult.OK)
            return;
        //Dacă starea anterioară a fost deja SingleFileState, doar actualizez
fișierul redat
        if (_context.StateNumber == MP3PlayerStates.SingleFileState)
        {
            ((AxWindowsMediaPlayer)_context.Controls[0]).URL =
Path.GetFullPath(openFileDialog.FileName);
        }
        //Altfel, schimb starea în SingleFileState și adaug controlul pe interfața
        else
        {

```

```

        //Elimin ce era pe vechea interfata, daca era un player pornit il
        opresc si creez un nou player pe care il
        //setez sa redea in bucla, cu dimensiunea si la locatia mentionate, si
        ii dau fisierul pe care sa-l redea
        groupBox.Controls.Clear();
        if (_context.StateNumber == MP3PlayerStates.PlaylistState ||
_context.StateNumber == MP3PlayerStates.RadioState)
            ((AxWindowsMediaPlayer)_context.Controls[0]).Ctlcontrols.stop();
        _context.StateNumber = MP3PlayerStates.SingleFileState;
        _context.Request();
        groupBox.Controls.Add(_context.Controls[0]);
        _context.Controls[0].CreateControl();
        ((AxWindowsMediaPlayer)_context.Controls[0]).settings.setMode("loop",
true);

        _context.Controls[0].Location = new System.Drawing.Point(6, 27);
        _context.Controls[0].Size = new System.Drawing.Size(498, 368);
        ((AxWindowsMediaPlayer)_context.Controls[0]).URL =
Path.GetFullPath(openFileDialog.FileName);
    }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
//Functie apelata cand se apasa "Deschidere playlist"
private void DeschiderePlaylistToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        //Cer utilizatorului sa deschida un playlist existent. Daca nu deschide un
        fisier valid, parasesc functia.
        openFileDialog.Filter = "Playlist(*.txt)|*.txt";
        if (openFileDialog.ShowDialog() != DialogResult.OK)
            return;
        List<string> melodii = new List<string>();
        StreamReader sr = new
StreamReader(Path.GetFullPath(openFileDialog.FileName));
        string[] lvls = sr.ReadToEnd().Split("\r\n".ToCharArray(),
StringSplitOptions.RemoveEmptyEntries);
        sr.Close();
        for (int i = 0; i < lvls.Length; i++)
            melodii.Add(lvls[i]);
        // Crearea listei cu perechi cheie-valoare (calea completa la fisier -
        numele fisierului)
        var files = melodii.Select(path => new { Path = path, FileName =
Path.GetFileName(path) }).ToList();
        //Daca starea anterioara a fost PlaylistState, doar reactualizez lista si
        redau noul prim fisier al playlist-ului.
        if (_context.StateNumber == MP3PlayerStates.PlaylistState)
        {
            // Setarea DataSource-ului pentru ListBox
            ((ListBox)_context.Controls[1]).DataSource = files;

            // Setarea DisplayMember-ului si ValueMember-ului pentru ListBox

```

```

        ((ListBox)_context.Controls[1]).DisplayMember = "FileName";
        ((ListBox)_context.Controls[1]).ValueMember = "Path";

        ((ListBox)_context.Controls[1]).SelectedIndexChanged +=
ListBoxPlaylist_SelectedIndexChanged;
        ((AxWindowsMediaPlayer)_context.Controls[0]).URL = ((dynamic)
((ListBox)_context.Controls[1]).SelectedItem).Path;
    }
    //Altfel, refac controalele aferente acestei stari si le afisez in
groupBox.
    else
    {
        groupBox.Controls.Clear();
        if (_context.StateNumber == MP3PlayerStates.SingleFileState ||
_context.StateNumber == MP3PlayerStates.RadioState)
            ((AxWindowsMediaPlayer)_context.Controls[0]).Ctlcontrols.stop();
        _context.StateNumber = MP3PlayerStates.PlaylistState;
        _context.Request();
        groupBox.Controls.Add(_context.Controls[0]);
        groupBox.Controls.Add(_context.Controls[1]);
        groupBox.Controls.Add(_context.Controls[2]);
        groupBox.Controls.Add(_context.Controls[3]);
        _context.Controls[0].Location = new System.Drawing.Point(6, 27);
        _context.Controls[0].Size = new System.Drawing.Size(498, 368);
        _context.Controls[1].Location = new System.Drawing.Point(531, 27);
        _context.Controls[1].Size = new System.Drawing.Size(200, 368);
        ((ListBox)_context.Controls[1]).HorizontalScrollbar = true;

        // Setarea DataSource-ului pentru ListBox
        ((ListBox)_context.Controls[1]).DataSource = files;

        // Setarea DisplayMember-ului si ValueMember-ului pentru ListBox
        ((ListBox)_context.Controls[1]).DisplayMember = "FileName";
        ((ListBox)_context.Controls[1]).ValueMember = "Path";

        ((ListBox)_context.Controls[1]).SelectedIndexChanged +=
ListBoxPlaylist_SelectedIndexChanged;
        ((AxWindowsMediaPlayer)_context.Controls[0]).URL = ((dynamic)
((ListBox)_context.Controls[1]).SelectedItem).Path;
        ((AxWindowsMediaPlayer)_context.Controls[0]).PlayStateChange +=
AxWindowsMediaPlayer_PlayStateChange;
        _context.Controls[2].Text = "Random";
        _context.Controls[2].Size = new System.Drawing.Size(66, 17);
        _context.Controls[2].Location = new System.Drawing.Point(740, 27);
        _context.Controls[3].Text = "Loop";
        ((CheckBox)_context.Controls[3]).CheckedChanged +=
PlaylistLoop_CheckedChanged;
        _context.Controls[3].Size = new System.Drawing.Size(66, 17);
        _context.Controls[3].Location = new System.Drawing.Point(740, 47);
    }
}
catch (Exception ex)
{

```

```

        MessageBox.Show(ex.Message);
    }
}
//Functie apelata atunci cand se apasa pe "Creare playlist"
private void CrearePlaylistToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        //Pentru aceasta stare, nu mai conteaza starea anterioara. Elimin toate
        elementele. Daca aveam player pornit, il opresc.
        //Aduug elementele la groupBox, dupa ce le-am creat in context si
        particularizat aici.
        groupBox.Controls.Clear();
        if (_context.StateNumber == MP3PlayerStates.SingleFileState ||
        _context.StateNumber == MP3PlayerStates.PlaylistState || _context.StateNumber ==
        MP3PlayerStates.RadioState)
        {
            ((AxWindowsMediaPlayer)_context.Controls[0]).Ctlcontrols.stop();
        }
        _context.StateNumber = MP3PlayerStates.MakePlaylistState;
        _context.Request();
        _context.Controls[0].Text = "Aduagă fișier";
        _context.Controls[1].Text = "Salvează playlist";
        _context.Controls[2].Enabled = false;
        _context.Controls[0].Location = new System.Drawing.Point(30, 30);
        _context.Controls[1].Location = new System.Drawing.Point(120, 30);
        _context.Controls[2].Location = new System.Drawing.Point(30, 60);
        ((TextBox)_context.Controls[2]).Multiline = true;
        ((TextBox)_context.Controls[2]).Size = new System.Drawing.Size(770, 300);
        ((Button)_context.Controls[0]).Click += AddButtonClickMakePlaylist;
        ((Button)_context.Controls[1]).Click += SaveButtonClickAddPlaylist;
        groupBox.Controls.Add(_context.Controls[0]);
        groupBox.Controls.Add(_context.Controls[1]);
        groupBox.Controls.Add(_context.Controls[2]);

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
//Functie apelata atunci cand se apasa pe meniul Modificare playlist existent
private void ModificarePlaylistExistentToolStripMenuItem_Click(object sender,
EventArgs e)
{
    try
    {
        //Indiferent de starea anterioara, elimin elementele din groupBox. Daca
        aveam player deschis, il opresc.
        //Schimb starea contextului si modific proprietatile noilor elemente, pe
        care le adaug la groupBox.
        groupBox.Controls.Clear();
        if (_context.StateNumber == MP3PlayerStates.SingleFileState ||
        _context.StateNumber == MP3PlayerStates.PlaylistState || _context.StateNumber ==
        MP3PlayerStates.RadioState)

```

```

        {
            ((AxWindowsMediaPlayer)_context.Controls[0]).Ctlcontrols.stop();
        }
        _context.StateNumber = MP3PlayerStates.EditPlaylistState;
        _context.Request();
        _context.Controls[0].Text = "Adaugare fisier";
        _context.Controls[0].Location = new System.Drawing.Point(420, 27);
        _context.Controls[0].Width = 150;
        ((Button)_context.Controls[0]).Click += AddButtonClickEditPlaylist;
        _context.Controls[1].Text = "Stergere index selectat";
        _context.Controls[1].Width = 150;
        _context.Controls[1].Location = new System.Drawing.Point(420, 67);
        ((Button)_context.Controls[1]).Click += RemoveButtonClickEditPlaylist;
        openFileDialog.Filter = "Playlist(*.txt)|*.txt";
        if (openFileDialog.ShowDialog() != DialogResult.OK)
            return;
        List<string> melodii = new List<string>();
        StreamReader sr = new
StreamReader(Path.GetFullPath(openFileDialog.FileName));
        string[] lvls = sr.ReadToEnd().Split("\r\n".ToCharArray(),
StringSplitOptions.RemoveEmptyEntries);
        sr.Close();
        for (int i = 0; i < lvls.Length; i++)
            melodii.Add(lvls[i]);
        var files = melodii.Select(path => new { Path = path, FileName =
Path.GetFileName(path) }).ToList();
        // Setarea DataSource-ului pentru ListBox
        ((ListBox)_context.Controls[2]).DataSource = files;

        // Setarea DisplayMember-ului si ValueMember-ului pentru ListBox
        ((ListBox)_context.Controls[2]).DisplayMember = "FileName";
        ((ListBox)_context.Controls[2]).ValueMember = "Path";
        _context.Controls[2].Size = new System.Drawing.Size(380, 368);
        _context.Controls[2].Location = new System.Drawing.Point(20, 27);

        _context.Controls[3].Text = "Salveaza playlist modificat";
        _context.Controls[3].Width = 150;
        _context.Controls[3].Location = new System.Drawing.Point(420, 107);
        ((Button)_context.Controls[3]).Click += SaveButtonClickEditPlaylist;
        groupBox.Controls.Add(_context.Controls[0]);
        groupBox.Controls.Add(_context.Controls[1]);
        groupBox.Controls.Add(_context.Controls[2]);
        groupBox.Controls.Add(_context.Controls[3]);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//Functie apelata atunci cand se apasa pe meniul de ascultare post radio
private void AscultarePostRadioToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        //Elimin elementele din groupBox. Daca aveam player deschis, il opresc.
    }
}

```



//Adaug noile elemente in context, apeland functia Request si le modific proprietatile aici.

```
groupBox.Controls.Clear();
if (_context.StateNumber == MP3PlayerStates.SingleFileState ||
_context.StateNumber == MP3PlayerStates.PlaylistState || _context.StateNumber ==
MP3PlayerStates.RadioState)
{
    ((AxWindowsMediaPlayer)_context.Controls[0]).Ctlcontrols.stop();
}
_context.StateNumber = MP3PlayerStates.RadioState;
_context.Request();
groupBox.Controls.Add(_context.Controls[0]);
groupBox.Controls.Add(_context.Controls[1]);
_context.Controls[0].Location = new System.Drawing.Point(6, 27);
_context.Controls[0].Size = new System.Drawing.Size(498, 368);
_context.Controls[1].Location = new System.Drawing.Point(531, 27);
_context.Controls[1].Size = new System.Drawing.Size(270, 368);
_context.Controls[1].Size = new System.Drawing.Size(270, 368);
var radioStations = new List<object>
{
    new { Name = "Europa FM", Url =
"https://astreaming.edi.ro:8443/EuropaFM_aac" },
    new { Name = "Kiss FM", Url =
"https://live.kissfm.ro:8443/kissfm.aacp" },
    new { Name = "Magic FM", Url =
"https://live.magicfm.ro:8443/magicfm.aacp" },
    new { Name = "Pro FM", Url =
"https://edge126.rcs-rds.ro/profm/profm.mp3" },
    new { Name = "Radio ZU", Url =
"http://zuicast.digitalag.ro:9420/zu" },
    new { Name = "Digi FM", Url =
"http://edge76.rdsnet.ro:84/digifm/digifm.mp3" },
    new { Name = "Virgin Radio", Url =
"https://astreaming.edi.ro:8443/VirginRadio_aac" }
};

var listBox = (ListBox)_context.Controls[1];
listBox.DataSource = radioStations;
listBox.DisplayMember = "Name";
((AxWindowsMediaPlayer)_context.Controls[0]).URL =
"https://astreaming.edi.ro:8443/EuropaFM_aac";
listBox.SelectedIndexChanged += ListBoxRadio_SelectedIndexChanged;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
```