



FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE DEPARTAMENTUL CALCULATOARE

## ZONGA: APICAȚITE MOBILĂ DE STREAMING BAZATĂ PE CLOUD

LUCRARE DE LICENȚĂ

Absolvent: Vlad TĂMAŞ

Coordonator științific: Şef lucr. ing. Cosmina IVAN

2014





#### FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE DEPARTAMENTUL CALCULATOARE

DECAN, **Prof. dr. ing. Liviu MICLEA**  DIRECTOR DEPARTAMENT, Prof. dr. ing. Rodica POTOLEA

Absolvent: Vlad TĂMAŞ

#### ZONGA PENTRU WINDOWS PHONE 8

- 1. **Enunțul temei:** Proiectul își propune realizarea unei aplicații mobile pentru sistemul de operare Windows Phone 8 al platformei Zonga. Principalele funcționalități sunt streaming de fișiere audio, redarea fișierelor audio în mod offline, vizualizarea recomandărilor, vizualizarea top-urilor din sistem, posibilitatea de a adăuga piese/albume/artiști la favorite.
- 2. **Conținutul lucrării:** Cuprins, Introducere, Obiectivele proiectului, Studiu bibliografic, Analiza și fundamentare teoretică, Proiectare în detaliu și implementare, Testare și validare, Manual de instalare și utilizare, Bibliografie, Anexe.
- 3. Locul documentării: Universitatea Tehnică din Cluj-Napoca, Departamentul Calculatoare
- 4. Data emiterii temei: 1 noiembrie 2013
- 5. Data predării: 3 Iulie 2014

Absolvent:

Coordonator stiințific:





#### FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE DEPARTAMENTUL CALCULATOARE

## Declarație pe proprie răspundere privind autenticitatea lucrării de licență

Subsemnatul(a)\_\_\_\_\_

legitimat(ă) cu \_\_\_\_\_\_ seria \_\_\_\_\_ nr. \_\_\_\_\_, autorul lucrării

	elaborată	în	vederea
susținerii exan	nenului de finalizare a studiilor de licență la Facultatea de	Auto	matică și
Calculatoare,	Specializarea		din
cadrul Univers	sității Tehnice din Cluj-Napoca, sesiunea		a anului
universitar	, declar pe proprie răspundere, că această lucrare	este 1	rezultatul
propriei activi	tăți intelectuale, pe baza cercetărilor mele și pe baza	. info	ormațiilor
obținute din su	rse care au fost citate, în textul lucrării, și în bibliografie.		

Declar, că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență.

In cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de licență*.

Data

Nume, Prenume

,

Semnătura

## Cuprins

Capito	olul 1	l. Introducere	1
1.1.	Con	textul proiectului	1
1.2.	Ceri	nțele proiectului	2
1.3.	Con	ținutul lucrării	2
Capito	olul 2	2. Obiectivele Proiectului	3
Capito	olul 3	3. Studiu Bibliografic	5
3.1.	Zong	ga API	5
3.2.	Siste	eme similare	6
3.2	2.1.	Spotify	6
3.2	2.2.	Pandora Radio	7
3.2	2.3.	Deezer	7
3.2	2.4.	Comparație între aceste aplicații și aplicația Zonga	7
Capito	olul 4	4. Analiză și Fundamentare Teoretică	9
4.1.	Ceri	nțele aplicației	9
4.1	.1.	Cerințe Funcționale ale aplicației	9
4.1	.2.	Cerințe Non-Funcționale ale aplicației	10
4.2.	Cazı	uri de utilizare	11
4.3.	Func	damentare teoretică	18
4.3	3.1.	Tehnologii și resurse utilizate	18
2	4.3.1.	1. Windows Phone 8	18
2	4.3.1.	2. Isolated Storage	21
2	4.3.1.	3. LINQ	23
2	4.3.1.	4. Servicii Web	24
2	4.3.1.	5. JSON.NET	26
4.3	3.2.	Design pattern-uri	27
2	4.3.2.	1. MVVM	27
2	4.3.2.	2. Abstract Factory	28
Capito	olul 5	5. Proiectare de Detaliu și Implementare	29
5.1.	Arhi	itectura sistemului Zonga	29
5.1	.1.	Rolurile componentelor sistemului	30
:	5.1.1.	1. Aplicațiile client	30
	5.1.1.	2. Aplicația server	30
5.2.	Arhi	itectura aplicației server	31
5.3.	Apli	cația Zonga pentru Windows Phone 8	32

5.3.1	. Utilitarul pentru request-uri	
5.3.2	. Nivelul de servicii	
5.3.3	. Nivelul de prezentare	
5.3.4	. Utilitarul pentru baza de date	
5.3.5	. Componenta pentru redarea melodiilor	
5.	3.5.1. Redarea melodiilor salvate local	35
5.	3.5.2. Redarea melodiilor de pe Internet	37
5.4. I	Detalii de implementare	
5.4.1	. Descărcarea playlist-urilor	
Capitol	ıl 6. Testare și Validare	40
Capitol	ıl 7. Manual de Instalare și Utilizare	44
7.1. I	nstalarea aplicației	44
7.1.1	. Instalarea din Windows Store	44
7.1.2	. Instalarea aplicație folosind unealta Application Deployment	44
7.2. N	Aanual de utilizare	45
7.2.1	. Pagina de autentificare și creare cont	45
7.2.2	. Pagina principală	45
7.2.3	. Pagina cu recomandări	46
7.2.4	. Pagina cu topuri	47
7.2.5	. Pagina cu elementele favorite	48
7.2.6	. Pagina playlist-urilor	48
7.2.7	. Pagina unui artist	49
7.2.8	. Pagina unui album	50
7.2.9	. Căutare	50
7.2.1	0. Quicklist	51
Capitol	ul 8. Concluzii și dezvoltări ulterioare	53
8.1. 0	Dbiective realizate	53
8.2. I	Promovarea aplicației	54
8.3. I	Dezvoltări ulterioare	54
Bibliog	afie	55
Glosar	le termeni	56
Listă ta	bele și figuri	56

## **Capitolul 1. Introducere**

Scopul acestei lucrări este crearea unui client mobil pentru sistemul de operare Windows Phone 8 a platformei Zonga. Zonga este cel mai mare serviciu local de muzică, lansat în 2012 de către echipa Trilulilu din dorința de a oferi tuturor românilor posibilitatea de a se bucura legal de muzica și artiștii lor favoriți, oricând și oriunde sar afla. Prin parteneriatele încheiate cu casele de discuri internaționale și locale, Zonga oferă acces legal la un catalog de peste 22 de milioane de melodii la cea mai înaltă calitate.

Cu ajutorul aplicației Zonga pentru Windows Phone 8 un utilizator poate asculta oricând online muzica preferată, poate descoperi artiști noi, melodii și albume, economisind astfel spațiul de stocare. De asemenea, în funcție de istoricul ascultărilor și de genurile preferate, utilizatorul va primi recomandări personalizate pentru a descoperi materiale noi. Catalogul este actualizat continuu cu cele mai noi melodii oficiale, de îndată ce acestea sunt disponibile pentru streaming.

#### 1.1. Contextul proiectului

Era dispozitivelor mobile a devansat cu mult apariția calculatoarelor în anii 1980 și apariția Internetului din anii 1990 în ceea ce privește adoptarea lor de către clienți. Până la sfârșitul anului 2014 eMarketer estimează că vor fi aproximativ 4.55 miliarde de dispozitive mobile active. Pentru a evidenția cât de rapidă a fost adopția telefoanelor mobile inteligente, este suficient să amintim că între 1981 și 2000 au fost vândute aproximativ 800 de milioane de calculatoare, ceea ce duce la un factor de adopție de 4 ori mai mare decât cel al adopției de calculatoare. Această creștere rapidă a pieței dispozitivelor mobile a creat noi posibilități pentru utilizatorii săi prin intermediul aplicațiilor din magazinele online.

Apariția și adopția rapida a dispozitivelor mobile și a Internetului a generat schimbări și în alte industrii precum cea muzicală. Dacă în urmă cu 10 ani utilizatorii cumpărau discuri cu melodiile formației lor preferate pentru a le asculta, acum aceștia pot să cumpere sau să descarce melodiile prin intermediul a câtorva click-uri. Datorită facilitaților oferite de Internet asistăm la o creștere drastică a colecțiilor de muzică pe care le deține fiecare utilizator în parte. Un utilizator sa dețină în colecția personala între 5000 și 10000 de melodii este ceva obișnuit în zilele noastre. Întreținerea unei asemenea colecții de melodii este o sarcină dificilă și în plus un număr atât de mare de fișiere audio poate ocupa mult spațiu. Problema spațiului de stocare apare în primul rând la dispozitivele mobile care au un spațiu de stocare mult mai mic decât un laptop sau un calculator.

O problemă în ziua de azi este portabilitatea. De cele mai multe ori oamenii doresc să aibă aceleași melodii pe telefonul mobil, pe tabletă, la birou sau acasă, iar pentru a avea sincronizata colecția de muzica trebuie să depună un efort suplimentar. Această problemă se rezolvă dacă utilizatorii au acces la melodiile lor preferate prin intermediul unei aplicații de streaming care să funcționeze pe orice dispozitiv mobil. O altă problemă este faptul că în lipsa Internetului utilizatorii nu mai au acces la muzică. Problema s-ar rezolva dacă aplicația cu ajutorul căreia utilizatorii ascultă melodiile pe Internet ar fi capabilă să salveze local o parte din melodii și să le poată reda chiar daca dispozitivul nu beneficiază de conexiune la Internet într-o anumită perioadă. O aplicație care rezolvă aceste probleme este aplicația Zonga pentru Windows Phone 8.

## **1.2.** Cerințele proiectului

Tema lucrării de fată o reprezintă crearea unei aplicații pentru sistemul de operare Windows Phone 8 care să se integreze cu platforma Zonga. Zonga este o platforma de muzică unde utilizatorii au acces pe baza de abonament lunar. Muzica este furnizată de către casele de discuri. Platforma are componente web, desktop si mobil (iPhone, Android, Symbian).

Folosind această aplicație utilizatorul va avea posibilitatea să asculte pe telefonul mobil direct de pe Internet sau salvată în memoria telefonului orice piesă dorește din cele peste 20 de milioane de melodii care există în cadrul platformei Zonga.

Un utilizator va putea de asemenea să vadă cele mai ascultate melodii, cei mai ascultați artiști și cele mai ascultate albume din cadrul platformei.

Fiecare utilizator va putea să vizualizeze recomandările melodiilor, artiștilor și albumelor pe baza a ceea ce a ascultat până în acel moment în cadrul platformei Zonga.

Pe lângă aceste funcționalități un utilizator își va putea crea propriile playlisturi pe care va putea să le asculte. Pentru ascultarea unui playlist un utilizator poate să asculte melodiile direct de pe Internet sau poate să descarce un playlist și să îl asculte atunci când nu are conexiune la Internet.

O altă funcționalitate importantă a aplicației este crearea de colecții de melodii, artiști sau albume favorite ale unui utilizator.

Platforma Zonga integrează și rețeaua de socializare Facebook. Când un utilizator ascultă cel puțin 30 de secunde dintr-o melodie, se va distribui toți prietenilor utilizatorului, din rețeaua de socializare, melodia pe care acesta o ascultă. Această opțiune de distribuire a melodiei este opțională și poate fi activată sau dezactivată din cadrul secțiunii de setări.

#### 1.3. Conținutul lucrării

În capitolul următor sunt prezentate obiectivele functionale care au dus la dezvoltarea produsului software și obiectivele de implementare care sunt necesare pentru integrarea datelor in aplicație într-un mod cat mai specific sistemului de operare Windows Phone 8. În capitolul 3 este prezentată metoda de integrare a aplicatiei cu platforma Zonga, controalele specifice Windows Phone 8 și vor fi descrise câteva aplicații similare, urmând ca apoi să fie prezentată o comparație intre aplicația curenta și aplicații similare identificate. În capitolul 4 sunt descrise mai întâi cerințele aplicației (cerintele functionale si cerintele non-functionale), vor fi identificati actorii si vor fi detaliate câteva cazuri de utilizare, după care se vor prezenta tehnologiile utilizate pentru dezvoltarea aplicației. În capitolul 5 se va prezenta arhitectura sistemului Zonga și componentele care îl alcătuiesc, apoi urmând să se descrie arhitectura aplicației, componentele sale si detalii de implementare mai complexe sau care au generat un se t de dificultăți. În capitolul 6 este prezentat modul de testare al aplicației și validarea rezultatelor obținute prin intermediul unor scenarii de testare. În capitolul 7 sunt descriși pașii de instalare a aplicației, iar apoi va fi întocmit un manual de utilizare. În capitolul 8 sunt subliniate câteva concluzii și câteva direcții de dezvoltare ulterioare a aplicației.

La final este adăugată o bibliografie și o anexă care conține un glosar de termeni, o listă cu figuri și o listă cu tabelele folosite în documentație.

## **Capitolul 2. Obiectivele Proiectului**

Precum s-a specificat anterior, proiectul de față are ca scop crearea și implementarea unei aplicații pentru dispozitive mobile care rulează pe platforma Windows Phone 8 și se integrează cu platforma de muzică Zonga. Zonga ajută utilizatorii să își gestioneze mai bine colecțiile personale de muzică legală oriunde și oricând.

Obiectivele acestui proiect se pot împărți în două categorii: obiective funcționale și obiective de implementare. Obiectivele de implementare se referă la studierea modului de integrare a aplicației cu platforma Zonga și la familiarizarea dezvoltatorului cu controalele specifice sistemului de operare Windows Phone 8. Obiectivele funcționale se referă la funcționalitățile care ar trebui să le aibă aplicația pentru a oferi utilizatorilor aceleași servicii ca și celelalte aplicații mobile ale platformei Zonga in contextul sistemului de operare Windows Phone 8. În continuare vor fi descrise obiectivele funcționale ale produsului software.

- Înregistrare. Utilizatorii trebuie să aibă posibilitatea să își creeze un cont pentru a beneficia de funcționalitățile platformei Zonga. Crearea unui cont se va putea face atât cu Facebook cât și cu o adresă de email. La crearea contului pe baza unei adrese de email se va verifica adresa de email, pentru a oferi o siguranță mai mare utilizatorilor. După verificarea adresei de email utilizatorii vor beneficia de planul tarifar FULL pentru o perioada de șapte zile de la crearea contului.
- Autentificare. Utilizatorii trebuie să se autentifice pentru a avea putea avea acces la funcționalitățile sistemului.
- **Căutare rapidă.** La introducerea unui text în interiorul unei casete de căutare utilizatorii vor primi sugestiile cele mai relevante pentru textul introdus. Sugestiile pot fi compuse din artiști, melodii sau albume ale unor artiști. Aceasta funcționalitate va îmbunătăți experiența utilizatorului.
- **Căutare avansată.** În cazul în care utilizatorul nu găsește ceea ce dorește să caute printre sugestiile de la căutarea rapidă, acesta poate să opteze pentru o căutare avansată. În cadrul căutării avansate utilizatorii vor vedea un număr mai mare de rezultate compuse din melodii, albume sau artiști decât în cazul căutării rapide.
- **Vizualizarea recomandărilor.** Utilizatorul va putea să vizualizeze recomandări de playlist-uri, artiști, melodii sau albume pe baza melodiilor ascultate până în momentul respectiv în cadrul platformei Zonga.
- **Vizualizarea topurilor.** Utilizatorii vor putea să vizualizeze topuri referitoare la cele mai ascultate melodii, cei mai ascultați artiști sau cele mai ascultate albume de către toți utilizatorii sistemului.
- **Operații pentru melodii/albume/artiști favorite.** Un utilizator va avea posibilitatea să își definească melodiile favorite, albumele favorite sau artiștii favoriți din cadrul sistemului.
- **Operații pentru playlist-uri.** Un utilizator va avea posibilitatea să își creeze playlist-uri personalizate.

- Salvarea unui playlist de la recomandări ca fiind al utilizatorului. Utilizatorii vor avea posibilitatea să salveze un playlist dintre cele recomandate pentru el ca aparținând utilizatorului.
- **Descărcarea unui playlist.** Utilizatorii aplicației vor avea posibilitatea să descarce un playlist în suportul de stocare al dispozitivului lor pentru a putea asculta melodii chiar și când nu are acces la Internet.
- **Redarea melodiilor.** Utilizatorii vor putea sa asculte peste 20 de milioane de melodii direct de pe Internet fără a fi nevoie să le descarce pe telefon. În cazul în care un utilizator nu are acces la Internet dar are un playlist descărcat, atunci acesta va putea să asculte melodiile din playlist-urile descărcate.

#### **Capitolul 3. Studiu Bibliografic**

În acest capitol vor fi prezentate principalele concepte relevante pentru dezvoltarea și implementarea cerințelor funcționale si non-funcționale ale proiectului cât și alte aplicații asemănătoare.

#### 3.1. Zonga API

Pentru integrarea conținutului în aplicație, Zonga pune la dispoziție un Application Programming Interface (API) bazat pe standardul JSON prin intermediul căruia se interacționează cu platforma Zonga. Acest API-ul poate fi accesat doar cu ajutorul unei chei private.

Elementele vizibile în aplicație se iau cu ajutorul unor cereri GET prin HTTPS către serverele Zonga la URL-ul https://api.zonga.ro/json?apikey=your\_api\_key, iar in corpul cereri se va pune un JSON de forma {"jsonrpc": "2.0", "method": "numele metodei", "params": [parametrii metodei], "id": "un id de request"}. În figura 3.1 este ilustrat un exemplu de cerere și un exemplu de răspuns pentru o căutare a artistului Voltaj, iar în cazul unei erori se v-a primi un răspuns JSON de forma celui ilustrat în figura 3.2.

```
{"jsonrpc": "2.0", "method": "search.results", "params": ["voltaj", 0, 2], "id": 45}
        "result" : {
                 "name"
                                  : "search:voltaj",
                 "title"
                                  : "voltaj",
                 "time"
                                  : "2010-11-15 12:20:42",
                 "count_entries" : 2,
                 "entries"
                                  : [
                         £
                                  "class"
                                                 : "play song",
                                  "identifier" : "ls663nt6",
                                                 : "Acolo",
                                  "title"
                                  "artist_name" : "voltaj",
"artist" : "Voltaj",
                                                   "Voltaj",
                                  "album_name" : "povestea-oricui",
                                  "album"
                                                 : "Povestea Oricui",
                                  "duration"
                                                 : 217,
                                  "favorite"
                                                 : false
                         },
                                  "class"
                                                 : "play song",
                                  "identifier" : "qi3l3kgn",
"title" : "Jur",
                                  "artist_name" : "voltaj",
                                  "artist"
                                                 : "Voltaj",
                                  "album_name" : "povestea-oricui",
                                  "album"
                                                 : "Povestea Oricui",
                                  "duration"
                                                : 245,
                                  "favorite"
                                                 : false
                         }
                1
        },
"id"
                  : "46",
        "error"
                  : null,
        "jsonrpc" : "2.0"
```

}

Figura 3.1 Exemplu de cerere și răspuns prin intermediul API-ului

```
{
    "result" : null,
    "error" : {
        "code" : -32000,
        "message" : "not_logged_in",
        "data" : {
        }},
    "id" : "47",
        "jsonrpc" : "2.0"
}
```

Figura 3.2 Exemplu de răspuns cu eroare

Toate funcționalitățile oferite de API sunt documentate la adresa https://api.zonga.ro/docs/zonga\_api/tutorial\_intro.pkg.html

#### **3.2.** Sisteme similare

Pentru a motiva crearea unei asemenea aplicații a fost nevoie de crearea unui studiu de piață. Acest studiu a urmărit un set de elemente cheie printre care se regăsesc numărul de potențiali utilizatori, numărul de dispozitive mobile cu Windows Phone 8, rata de adopție a telefoanelor cu acest sistem de operare și aplicațiile de streaming existente pe piața internațională. A fost nevoie de efectuarea unei comparații între aplicațiile existente și aplicația care urmează să fie dezvoltată pentru a scoate în evidență atuurile aplicației care urmează să fie dezvoltată.

Au fost alese aplicațiile descrise mai jos deoarece se apropie ca funcționalități de aplicația Zonga pentru Windows Phone 8. Principalul argument pentru care au fost selectate aceste aplicații este faptul că oferă servicii pe mai multe sisteme de operare.

#### 3.2.1. Spotify

Sporify este un serviciu de streaming de muzică comercială oferind conținut audio cu drepturi de gestionare restricționat de la case de discuri precum Sony, EMI, Warner Music Group si Universal Music Group. Aplicația oferă posibilitatea de a accesa melodii, albume, artiști, genuri, plalisty-uri sau case de discuri. Abonamentele "Premium" permit utilizatorilor să descarce muzica pentru a putea fi ascultată offline și elimină reclamele.

Spotify este disponibil în aproximativ 20 de țări, dintre care lipsește Romania, și în peste 50 de limbi. Catalogul său de muzică numără aproximativ 20 de milioane de melodii care pot fi accesate prin intermediul artiștilor, albumelor, caselor de discuri și a genurilor muzicale. Unii artiști pot să lipsească din catalogul Spotify în anumite regiuni din cauza drepturilor de autor.

Utilizatorii pot sa își creeze playlist-uri care mai apoi pot să le împartă cu prietenii și împreună cu aceștia pot să le gestioneze. Playlist-urile descărcate se sincronizează automat în cazul în care alt utilizator șterge sau adaugă o melodie la playlist.

O altă funcționalitate este radioul care este disponibil atât pentru utilizatorii posesori se conturi gratuite cât și pentru cei de conturi Premium. Radioul creează o listă cu melodii alese la întâmplare în funcție de tipul de radio pe care utilizatorul dorește să îl asculte. De exemplu dacă utilizatorul dorește să asculte radioul unui artist, acesta va asculta un playlist cu melodii alese la întâmplare de la artistul respectiv.

O funcționalitate la care Sptofy este deficitar este oferirea de recomandări pentru playlist-uri, melodii, artiști sau albume, însă se poate integra cu serviciul platforma Last.fm care oferă aceste funcționalități.

Spotify permite utilizatorilor să își creeze conturi cu ajutorul platformelor de socializare precum Facebook si Twitter. Odată ce un utilizator își integrează contul de Spotify cu unul de socializare, acesta poate să acceseze muzica favorită a prietenilor săi și playlist-urile acestora. În momentul în care un utilizator ascultă o melodie și are integrat de exemplu contul de Facebook, prietenii de pe rețeaua de socializare vor vedea în news feed-ul lor ce melodie ascultă utilizatorul respectiv.

#### **3.2.2. Pandora Radio**

Pandora este un serviciu de streaming de muzică și generare automată de recomandări. Serviciul este disponibil doar in Statele Unite, Australia si Noua Zeelandă. Pandora redă melodiile de la un anumit gen pe baza artiștilor selectați de către utilizator. După redarea melodiei, utilizator clasifică acea melodie ca fiind bună sau nu, iar serviciul ia in considerare această părere pentru selectarea melodiilor viitoare. În această selecție intră aproximativ 400 de atribute de care se ține cont. Atributele sunt grupate în grupuri mai mari numite grupuri concentrate cu trăsături comune.

Pandora oferă utilizatorilor posibilitatea sa cumpere melodiile sau albumele pe ca re le ascultă de la diverse magazine online.

#### **3.2.3.** Deezer

Deezer este un serviciu de streaming de muzică care permite utilizatorilor săi să asculte melodii de la numeroase case de discuri precum Sony, EMI, Warner Musci Group și Universal Music Group pe diferite dispozitive mobile în mod online sau offline. Deezer are aproximativ 30 de milioane de melodii licențiate și 30.000 de stații de radio în catalogul său. Din toate aceste melodii unele e posibil să nu fie disponibile în anumite regiuni ale lumii din cauza drepturilor de autor.

Utilizatorii pot să își creeze playlist-uri pe care mai apoi să le împărtășească cu prietenii și să le gestioneze împreună cu aceștia. Playlist-urile se pot descărca ca mai apoi să poată fi ascultate în mod offline.

La capitolul recomandări Deezer este și deficitar, însă acest neajuns poate fi recompensat de faptul că se poate integra cu Last.fm.

O altă funcționalitate este SmartRadio care este un radio bazat pe un artist și care generează un playlist cu melodii aleatoare de la artistul respectiv. Pe lângă acest radio utilizatorii au posibilitatea să asculte din cele 30.000 de radiouri cu teme diferite.

Deezer permite utilizatorilor să își creeze conturi cu ajutorul platformelor de socializare precum Facebook si Twitter. Odată ce un utilizator își integrează contul de Deezer cu unul de socializare, acesta poate să acceseze muzica favorită a prietenilor săi și playlist-urile acestora. În momentul în care un utilizator ascultă o melodie și are integrat de exemplu contul de Facebook, prietenii de pe rețeaua de socializare vor vedea în news feed-ul lor ce melodie ascultă utilizatorul respectiv.

Servicul Deezer permite utilizatorilor să încarce propriile melodii MP3 pe serverele sale. Într-un număr nelimitat.

Pe lângă funcționalitățile prezentate mai sus, Deezer oferă utilizatorilor posibilitatea de a vedea ultimele noutăți din cadrul sistemului.

## 3.2.4. Comparație între aceste aplicații și aplicația Zonga

Atât aplicația care se dorește a fi implementată, cât și aplicațiile prezentate mai sus au rolul de a face streaming la melodiile din cadrul platformei respective.

Avantajul principal al aplicației este disponibilitatea sa în Romania. Celelalte aplicații cu excepția Deezer nu sunt disponibile în Romania.

Sistemul Zonga, din punct de vedere al catalogului de melodii, se situează pe locul doi după Deezer și înainte de Spotify. Pandora nu poate intra în calcul la acest criteriu de comparație deoarece sistemul este proiectat ca un radio online și nu are un număr limitat de piese. O funcționalitate importantă este afișarea noutăților în cadrul aplicației deoarece utilizatorii vor fi mereu interesați de ultimele noutăți în materie de muzică pe care să o asculte. Această funcționalitate este oferită doar de aplicația Zonga și Deezer.

Crearea de playlist-uri și gestionarea acestora poate fi efectuată în aplicațiile Zonga, Spotify și Deezer. Împărțirea unui playlist cu prietenii de pe rețelele de socializare Facebook și Twitter se poate face de asemenea în toate cele, însă în cadrul aplicației Zonga nu se poate face gestionarea acestora de către prieteni.

Vizualizarea celor mai ascultate melodii, celor mai ascultate albume și cei mai ascultați artiști nu este disponibilă în cadrul aplicației Pandora, în rest toate aplicațiile au această funcționalitate.

În ceea ce privește recomandarea de melodii, albume și artiști se pot diferenția două categorii și anume: cele care au un sistem de recomandare customizat și cele care se folosesc de alte servicii pentru a oferi acest lucru. În prima categorie intră Zonga și Pandora, iar în ultima intra Spotify și Deezer care utilizează Last.fm pentru recomandarea melodiilor.

Din punct de vedere al radioului versiunea curentă a aplicației Zonga nu suportă nici un fel de radio chiar dacă în cadrul platformei Zonga există această funcționalitate. Cel mai complexă aplicație la această funcționalitate este aplicația Pandora deoarece este specializată pe acest segment. Deezer și Spotify oferă și ele această funcționalitate.

Căutarea după anumite cuvinte cheie este posibilă în cadrul aplicațiilor Zonga, Spotify și Deezer.

O funcționalitate la care Zonga nu are rival sunt elementele favorite ale unui utilizator. Nici o altă aplicație nu oferă această funcționalitate.

Redarea melodiilor se poate face atât în mod offline cât și în mod online la aplicațiile Zonga și Deezer, iar la aplicațiile Pandora și Spotify redarea se poate face doar în mod online.

Criteriul	Zonga	Spotify	Deezer	Pandora
Topuri din cadrul platformei	DA	DA	DA	NU
Sistem de recomandare nativ	DA	NU	NU	DA
Integrare cu sisteme de recomandare	NU	DA	DA	NU
externe (Last.fm)				
Radio	NU	DA	DA	DA
Categoria favorite	DA	NU	NU	NU
Căutare	DA	DA	DA	NU
Mod offline	DA	NU	DA	NU
Noutăți în cadrul sistemului	DA	NU	DA	NU
Integrare cu rețele de socializare	DA	DA	DA	NU
Gestionarea playlist-urilor distribuită	NU	DA	DA	NU
Disponibil în România	DA	NU	DA	NU
Are aplicație Windows Phone 8	NU	DA	DA	NU

Am surprins această comparație în tabelul de mai jos:

Tabel 3.1 Comparație între aplicațiile descrise și aplicația curentă

Pe baza comparației de mai sus se poate observa necesitatea dezvoltării unei aplicații pe platforma Windows Phone 8 pentru sistemul Zonga.

## Capitolul 4. Analiză și Fundamentare Teoretică

In acest capitol se va fi realizata o analiza a aplicației, care va permite observarea modului de funcționare a sistemului și ce funcționalități de baza ar trebui să ofere. De asemenea, vor fi descrise tehnologiile care se vor utiliza pentru realizarea aplicației.

## 4.1. Cerințele aplicației

Cerințele aplicației sunt descrieri ale serviciilor oferite de sistem și ale constrângerilor care sunt generate de-a lungul desfășurării procesului de inginerie a cerințelor. Acestea pornesc de la afirmații abstracte de nivel înalt și se detaliază până la specificații matematice funcționale.

Cerințele care sunt evidențiate în această secțiune sunt cele pentru dezvoltarea de aplicații bazate pe streaming și gestiunea colecțiilor mari de muzică. O cerință ar fi descrierea a ceea ce face concret aplicația și modul în care este atins obiectivul respectiv.

Cerințele care trebuie îndeplinite pentru a asigura bună funcționare a aplicației vor fi descrise în cadrul Cerințelor Funcționale. Acestea definesc capacitatea sistemelor și a serviciilor.

În cadrul Cerințelor Non-Funcționale se vor descrie constrângerile referitoare la calitatea sistemului, prin identificarea atributelor sistemului. Acestea descriu nivelul de calitate dorit, de exemplu "Cât de securizată este aplicația?" sau "Cât de ușor este de folosit". Constrângerile sunt condiții non-negociabile care afectează sistemul. O cerință non-funcțională este o cerință care specifică criteriul ce poate fi folosit la aprecierea unui sistem, mai degrabă decât comportamente specifice.

## 4.1.1. Cerințe Funcționale ale aplicației

Cerințele funcționale descriu funcțiile pe care trebuie să le realizeze sistemul și serviciile oferite. Se menționează ce transformări trebuie efectuate asupra intrărilor și ce ieșiri trebuie să se obțină pentru fiecare tip de intrare.

O funcție este descrisă că un set de intrări și ieșiri. Cerințele funcționale pot fi calcule, detalii tehnice, manipulări și procesări de date, dar și alte funcționalități specifice care definesc ce trebuie să îndeplinească un sistem, care este scopul său.

În tabelul 4.1 sunt descrise principalele caracteristici funcționale pe care trebuie să le îndeplinească aplicația Zonga pentru Windows Phone 8.

<b>CF 1</b>	Vizualizarea topurilor	
CF 1.1	Vizualizarea topurilor de melodii	
CF 1.2	Vizualizarea topurilor de artiști	
CF 1.3	Vizualizarea topurilor de albume	
<b>CF 2</b>	Vizualizarea recomandărilor si gestionarea playlist-urilor	
	recomandate	
CF 2.1	Vizualizarea melodiilor recomandate	
CF 2.2	Vizualizarea artiștilor recomandați	
CF 2.3	Vizualizarea albumelor recomandate	
CF 2.4	Vizualizarea playlist-urilor recomandate	
CF 2.5	Salvarea unui playlist recomandat în colecția de playlist-uri	
CF 3	Vizualizarea și gestionarea favoritelor	

CF 3.1	Vizualizarea melodiilor favorite
CF 3.2	Vizualizarea artiștilor favoriți
CF 3.3	Vizualizarea albumelor favorite
CF 3.4	Adăugarea sau ștergerea unei melodii la favorite
CF 3.5	Adăugarea sau ștergerea unor artiști la favorite
CF 3.6	Adăugarea sau ștergerea unui album la favorite
<b>CF 4</b>	Căutare
CF 4.1	Oferire de sugestii
CF 4.2	Căutare melodii
CF 4.3	Căutare artiști
CF 4.4	Căutare albume
<b>CF 5</b>	Vizualizarea noutăților
CF 5.1	Vizualizarea melodiilor noi
CF 5.2	Vizualizarea albumelor noi
CF 5.3	Vizualizarea artiștilor noi
<b>CF 6</b>	Vizualizarea și gestionarea playlist-urilor personale
CF 6.1	Crearea unui playlist
CF 6.2	Redenumirea unui playlist
CF 6.3	Ştergerea unui playlist
CF 6.4	Adăugarea unei melodii într-un playlist
CF 6.5	Ştergerea unei melodii dintr-un playlist
CF 6.6	Descărcarea unui playlist
<b>CF 7</b>	Redarea melodiilor de pe Internet sau de pe disk
CF 7.1	Redarea melodiilor din quicklist în mod consecutiv
CF 7.2	Redarea melodiilor din quicklist în mod aleator
CF 7.3	Redarea unei melodii din quicklist de un număr infinit de ori
CF 7.4	Redarea melodiilor quicklist-ului în mod repetat
<b>CF 8</b>	Gestionarea qucicklist-ului
CF 8.1	Adăugarea unei melodii
CF 8.2	Ștergerea unei melodii
CF 8.3	Adăugarea unui album
CF 8.4	Adăugarea unui playlist
<b>CF 9</b>	Gestionarea setărilor

## 4.1.2. Cerințe Non-Funcționale ale aplicației

Cerințele non-funcționale sunt atașate cerințelor funcționale, dar sunt mai critice decât acestea. Dacă nu sunt îndeplinite, există posibilitatea că sistemul să nu mai îndeplinească scopul pentru care a fost creat.

Cerințele non-funcționale sunt calități și proprietăți pe care aplicația trebuie să le posede în timp ce îndeplinește cerințele funcționale pentru care a fost gândit.

Utilizatorii au așteptări mari în legătură cu funcționalitatea aplicației. Printre aceste așteptări se numără și "cât este de ușor de folosit aplicația, cât de rapid răspunde la comenzi, care este nivelul de încredere sau cât de bine se comportă în situații neașteptate". Cerințele non-funcționale definesc aceste aspecte.

Spre deosebire de cerințele funcționale, care definesc un comportament specific al funcțiilor, cerințele non-funcționale specifică criteriul ce poate fi folosit la aprecierea operațiilor unei aplicații.

Un exemplu de cerință non-funcțională sunt calitățile aplicației, acestea putând fi împărțite în două categorii principale: calități de execuție (securitate și utilizabilitate) și calități de evoluție (testabilitate, mentenabilitate, etc.).

În tabelul 4.2 se prezintă principalele caracteristici non-funcționale ale aplicație Zonga pentru Windows Phone 8.

CNF 1 -	Aplicația trebuie să utilizeze într-un mod cât mai eficient posibil	
Eficiența	traficul de date cu serverul sistemului. Pentru un consum cât mai	
	redus de energie trebuie să se aibă în vedere utilizarea memoriei	
	telefonului și a procesorului. În momentul în care se ascultă o	
	melodie, toate resursele utilizate pentru partea de redare nu	
	trebuie sa depășească 15Mb de memorie.	
CNF 2 –	Aplicația este conceputa într-o manieră ce presupune adăugarea	
Flexibilitate	cu ușurința a unor funcționalități noi.	
CNF 3 –	Acces restrâns la aplicație, imposibil de utilizat fără	
Integritate	autentificarea unui utilizator cu un nume și o parola.	
CNF 4 –	Aplicația folosește multe patten-uri și arhitectura de tip N-	
Reutilizabilitate	Layer, ceea ce o face ușor de separat pe componente care pot fi	
	reutilizate în cadrul altor aplicați.	
CNF 5 –	Aplicația este testată împotriva anumitor erori. Va tolera date	
Robustețe	invalide însă nu se va bloca sau se va opri din funcționare.	
CNF 6 –	A pliantia are a interfată pristaneagă, vear de înteleg și falogit	
Utilizabilitate	Apricăția are o internăță prietenoasă, ușor de ințeles și folosit.	
CNF 7 –	Aplicația trebuie să satisfacă anumite reguli de autorizare și	
Securitate	autentificare pentru a se integra cu sistemul Zonga.	
CNF 8 –	Aplicatio trabuja să fia usar de întratinut	
Mentenabilitate	Apricaçãa neoure sa ne uşor de inneçindi.	

Tabel 4.2 Cerințe Non-Funcționale

## 4.2. Cazuri de utilizare

Diagramele cazurilor de utilizare descriu comportamentul sistemului, oferind o imagine de ansamblu asupra modului în care acesta este folosit din punct de vedere al utilizatorilor. Cazurile de utilizare realizează o descriere, din punct de vedere funcțional, a aplicației. Ansamblul tuturor cazurilor de utilizare și utilizatorii acestora (actorii) formând modelul cazurilor de utilizare.

Aplicația Zonga pentru Windows Phone 8 are un singur actor care are acces la toate funcționalitățile aplicației.

În continuare se va realiza o scurtă descriere a celor mai importante cazuri de utilizare, care vor fi grupate în funcție de cerințele funcționale ale aplicației. O cerința funcționala va fi însoțită și de o diagramă a cazurilor de utilizare.

Prima cerința funcțională este descrisă în cazul de utilizare CU 1 și este ilustrată în figura 4.8.



Figura 4.1 Diagrama Use Case pentru CF 1

## CU 1. Titlu: Vizualizarea topurilor

#### **Descriere:**

Acest scenariu descrie pașii pe care trebuie să îi facă utilizatorul autentificat care dorește să vizualizeze cele mai ascultate melodii, cei mai ascultați artiști sau cele mai ascultate albume de-a lungul timpului în cadrul platformei Zonga.

## Actor primar: Utilizator simplu

Precondiții: Utilizatorul trebuie să fie autentificat

## Post condiții: Nu Există

#### Scenariu de succes:

- 1. Navighează la pagina cu topuri.
- 2. Aplicația verifica conexiunea la Internet. Aceasta există.

3. Apar primele 20 cele mai ascultate melodii.

4. Când utilizatorul se apropie de sfârșitul listei cu melodii, aplicația va încărca următoarele 20 de melodii din top.

5. Swipe stânga. Se vor încărca primele 20 cele mai ascultate albume.

6. Când utilizatorul se apropie de sfârșitul listei cu albume, aplicația va încărca următoarele 20 de albume din top.

7. Swipe stânga. Se vor încărca primi 20 cei mai ascultați artiști.

8. Când utilizatorul se apropie de sfârșitul listei cu artiști, aplicația va încărca următorii 20 de artiști din top.

#### Scenariu de eșec 1:

1. Navighează la pagina cu topuri

2. Aplicația verifica conexiunea la Internet. Aceasta nu există.

3. Utilizatorul va fi notificat printr-un mesaj de eroare, iar vizualizarea topurilor nu mai este posibilă.

## Scenariu de eşec 2:

1. Utilizatorul încărcă elemente din cadrul topurilor pană când aplicația se apropie de pragul maxim de memorie pe care îl poate utiliza.

2. Utilizatorul va fi notificat printr-un mesaj de eroare, iar încărcarea altor elemente în cadrul secțiunii topuri nu va mai fi posibilă.

A doua cerința funcțională este descrisă în cazul de utilizare CU 2 și este ilustrată în figura 4.9.

#### CU 2. Titlu: Vizualizarea recomandărilor Descriere:

Acest scenariu descrie pașii pe care trebuie să îi facă utilizatorul autentificat care dorește să vizualizeze recomandările oferite de către sistemul Zonga pe baza istoricului ascultărilor și a genurilor favorite.

Actor primar: Utilizator simplu

Precondiții: Utilizatorul trebuie să fie autentificat

Post condiții: Nu Există

#### Scenariu de succes:

1. Navighează la pagina cu recomandări.

2. Aplicația verifică conexiunea la Internet. Aceasta există.

3. Apar primele 20 de melodii recomandate.

4. Când utilizatorul se apropie de sfârșitul listei cu melodii, aplicația va încărca următoarele 20 de melodii recomandate.

5. Swipe stânga. Se vor încărca primele 20 de albume recomandate.

6. Când utilizatorul se apropie de sfârșitul listei cu albume, aplicația va încărca următoarele 20 de albume recomandate.

7. Swipe stânga. Se vor încărca primi 20 de artiști recomandați.

8. Când utilizatorul se apropie de sfârșitul listei cu artiști, aplicația va încărca următorii 20 de artiști recomandați.

9. Swipe stânga. Se vor încărca primele 20 de playlist-uri recomandate.

10. Când utilizatorul se apropie de sfârșitul listei cu playlist-uri, aplicația va încărca următoarele 20 de playlist-uri recomandate.

#### Scenariu de eșec 1:

1. Navighează la pagina de recomandari

2. Aplicația verifică conexiunea la Internet. Aceasta nu există.

3. Utilizatorul va fi notificat printr-un mesaj de eroare, iar vizualizarea recomandărilor nu mai este posibilă.

#### Scenariu de eșec 2:

1. Utilizatorul încărca elemente din cadrul recomandărilor până când aplicația se apropie de pragul maxim de memorie pe care o poate utiliza.

2. Utilizatorul va fi notificat printr-un mesaj de eroare, iar încărcarea altor elemente în cadrul secțiunii recomandări nu va mai fi posibilă.



Figura 4.2 Diagrama Use Case pentru CF 2



A treia cerința funcționala este descrisa in cazurile de utilizare CU 3, CU 4, CU 5, CU 6 și este ilustrată în figura 4.10

Figura 4.3 Diagrama Use Case pentru CF 3

## CU 3. Titlu: Vizualizarea favoritelor

#### **Descriere:**

Acest scenariu descrie pașii pe care trebuie să îi facă utilizatorul autentificat care dorește să vizualizeze elementele sale favorite în cadrul sistemul Zonga.

## Actor primar: Utilizator simplu

Precondiții: Utilizatorul trebuie sa fie autentificat

## Post condiții: Nu Există

## Scenariu de succes:

- 1. Navighează la pagina cu favoritelor.
- 2. Aplicația verifica conexiunea la Internet. Aceasta există.
- 3. Apar primele 20 de melodii favorite.

4. Când utilizatorul se apropie de sfârșitul listei cu melodii, aplicația va încărca următoarele 20 de melodii favorite.

5. Swipe stânga. Se vor încărca primele 20 de albume favorite.

6. Când utilizatorul se apropie de sfârșitul listei cu albume, aplicația va încărca următoarele 20 de albume favorite.

7. Swipe stânga. Se vor încărca primi 20 de artiști favoriți.

8. Când utilizatorul se apropie de sfârșitul listei cu artiști, aplicația va încărca următorii 20 de artiști favoriți.

#### Scenariu de eșec 1:

1. Navighează la pagina favoritelor

2. Aplicația verifica conexiunea la Internet. Aceasta nu există.

3. Utilizatorul va fi notificat printr-un mesaj de eroare, iar vizualizarea favoritelor nu mai este posibilă.

## Scenariu de eșec 2:

1. Utilizatorul încărca elemente din cadrul favoritelor până când aplicația se apropie de pragul maxim de memorie pe care o poate utiliza.

2. Utilizatorul va fi notificat printr-un mesaj de eroare, iar încărcarea altor elemente în cadrul secțiunii favorite nu va mai fi posibilă.

# **CU 4. Titlu:** *Adăugarea/ștergerea unei melodii de la favorite* **Descriere:**

Acest scenariu descrie pașii pe care trebuie sa îi facă utilizatorul autentificat care dorește să adauge sau să șteargă melodii ca fiind favorite în cadrul sistemul Zonga. **Actor primar:** Utilizator simplu

Precondiții: Utilizatorul trebuie să fie autentificat

Post condiții: Nu Există

## Scenariu de succes:

- 1. Navighează la orice pagină din aplicație care conține o listă de melodii.
- 2. Aplicația verifica conexiunea la Internet. Aceasta exista.
- 3. Apar primele 20 de melodii din pagina respectiva.
- 4. Utilizatorul execută acțiunea "tap and hold" pentru afișarea unui meniu.

5. Dacă melodia nu este adăugată la favorite printre opțiunile meniului este și "adaugă la favorite". Utilizatorul dă tap pe opțiune, iar melodia va fi adăugată la favorite. Dacă melodia este adăugată la favorite, printre opțiunile meniului se regăsește "șterge de la favorite". Utilizatorul dă tap pe opțiune iar melodia va fi ștearsa de la favorite

#### Scenariu de eșec:

1. Navighează la orice pagină din aplicație care conține o lista de melodii.

2. Încearcă să se adauge/șteargă o melodie de la favorite, iar conexiunea la Internet se întrerupe.

3. Utilizatorul va fi notificat printr-un mesaj de eroare, iar adăugarea la favorite nu mai este posibilă.

## **CU 5. Titlu:** *Adăugarea/ștergerea unui album de la favorite* **Descriere:**

Acest scenariu descrie pașii pe care trebuie să îi facă utilizatorul autentificat care dorește să adauge sau să șteargă albume ca fiind favorite în cadrul sistemul Zonga. **Actor primar:** Utilizator simplu

Precondiții: Utilizatorul trebuie să fie autentificat

## Post condiții: Nu Există

## Scenariu de succes:

- 1. Navighează la orice pagină din aplicație care conține o lista de albume.
- 2. Aplicația verifica conexiunea la Internet. Aceasta există.
- 3. Apar primele 20 de albume din pagina respectivă.
- 4. Utilizatorul execută acțiunea "tap and hold" pentru afișarea unui meniu.

5. Dacă albumul nu este adăugat la favorite, printre opțiunile meniului este și "adaugă la favorite". Utilizatorul dă tap pe opțiune iar albumul va fi adăugat la favorite. Dacă albumul este adăugat la favorite, printre opțiunile meniului se regăsește "șterge de la favorite". Utilizatorul dă tap pe opțiune iar albumul va fi șters de la favorite

## Scenariu de eșec:

1. Navighează la orice pagină din aplicație care conține o lista de albume.

2. Încearcă să se adauge/șteargă un album de la favorite, iar conexiunea la Internet se întrerupe.

3. Utilizatorul va fi notificat printr-un mesaj de eroare, iar adăugarea la favorite nu mai este posibilă.

# **CU 6. Titlu:** *Adăugarea/ștergerea unui artist de la favorite* **Descriere:**

Acest scenariu descrie pașii pe care trebuie să îi facă utilizatorul autentificat care dorește să adauge sau să șteargă artiști ca fiind favoriți în cadrul sistemul Zonga. **Actor primar:** Utilizator simplu

Precondiții: Utilizatorul trebuie să fie autentificat

Post condiții: Nu Există

#### Scenariu de succes:

- 1. Navighează la orice pagină din aplicație care conține o listă de artiști.
- 2. Aplicația verifică conexiunea la Internet. Aceasta există.
- 3. Apar primi 20 de artiști din pagina respectivă.
- 4. Utilizatorul execută acțiunea "tap and hold" pentru afișarea unui meniu.

5. Dacă artistul nu este adăugat la favorite, printre opțiunile meniului este și "adăuga la favorite". Utilizatorul dă tap pe opțiune, iar artistul va fi adăugat la favorite. Dacă artistul este adăugat la favorite, printre opțiunile meniului se regăsește "șterge de la favorite". Utilizatorul dă tap pe opțiune iar artistul va fi șters de la favorite

#### Scenariu de eșec:

1. Navighează la orice pagină din aplicație care conține o listă de artiști.

2. Încearcă să se adauge/șteargă un artist de la favorite, iar conexiunea la Internet se întrerupe.

3. Utilizatorul va fi notificat printr-un mesaj de eroare, iar adăugarea la favorite nu mai este posibilă.

În continuare va fi descris modelul de use case pentru cerința funcționala 4 și va fi reprezentată printr-o diagramă de use case așa cum apare in imaginea 4.11



Figura 4.4 Diagrama Use Case pentru CF 4

## CU 7. Titlu: Căutare

#### **Descriere:**

Acest scenariu descrie pașii pe care trebuie să îi facă utilizatorul autentificat care dorește să caute melodii, artiști sau albume în cadrul sistemul Zonga.

Actor primar: Utilizator simplu

Precondiții: Utilizatorul trebuie să fie autentificat

## Post condiții: Nu Există

#### Scenariu de succes:

1. Navighează la pagina principală.

2. Introdu caractere în caseta de text de deasupra meniului principal.

2. Aplicația verifică conexiunea la Internet. Aceasta există.

3. Pe baza textului introdus aplicația oferă sugestii care conțin melodii, albume și artiști.

4. Pentru o căutare mai detaliată apasă butonul de căutare.

5. Utilizatorul este dus la pagina de căutare avansată în care va vede prima dată o listă de melodii.

6. Când utilizatorul se apropie de sfârșitul listei cu melodii, aplicația va încărca următoarele 20 de melodii care se potrivesc criteriului de căutare.

5. Swipe stânga. Se vor încărca primele 20 de albume care se potrivesc cu criteriul de căutare.

7. Când utilizatorul se apropie de sfârșitul listei cu albume, aplicația va încărca următoarele 20 de albume care se potrivesc criteriului de căutare.

8. Swipe stânga. Se vor încărca primi 20 de artiști care se potrivesc criteriului de căutare.

9. Când utilizatorul se apropie de sfârșitul listei cu artiști, aplicația va încărca următorii 20 de artiști care se potrivesc criteriului de căutare.

#### Scenariu de eșec:

1. Navighează la pagina de principală

2. Aplicația verifică conexiunea la Internet. Aceasta nu există.

3. Utilizatorul va fi notificat printr-un mesaj de eroare, iar vizualizarea căutării nu mai este posibilă.

Cazul de utilizare numărul 8 descrie in detaliu cerința funcțională 5 și este reprezentată prin diagrama use case ilustrată în figura 4.12



Figura 4.5 Diagrama Use Case pentru CF 5

## CU 8. Titlu: Vizualizare noutăți

## **Descriere:**

Acest scenariu descrie pașii pe care trebuie să îi facă utilizatorul autentificat care dorește să vizualizeze noutățile din cadrul platformei Zonga.

Actor primar: Utilizator simplu

Precondiții: Utilizatorul trebuie să fie autentificat

Post condiții: Nu Există

Scenariu de succes:

1. Navighează la pagina principală.

2. Swipe dreapta

3. Aplicația verifică conexiunea la Internet. Aceasta există.

4. Aplicația încarcă cele mai noi melodii, albume și artiști din cadrul platformei Zonga.

#### Scenariu de eșec:

1. Navighează la pagina de principală

2. Swipe dreapta.

3. Aplicația verifică conexiunea la Internet. Aceasta nu există.

4. Utilizatorul va fi notificat printr-un mesaj de eroare, iar vizualizarea căutării nu mai este posibilă.

## 4.3. Fundamentare teoretică

Acest subcapitol prezintă o sinteză teoretică a tehnologiilor precum și a design pattern-urilor folosite. Astfel se prezintă tehnologiile alese și folosite pentru dezvoltarea sistemului atât la nivel de structura a aplicației, comunicare cu servicii Web cât și la nivelul interfeței și legăturilor dintre ele.

## 4.3.1. Tehnologii și resurse utilizate

## 4.3.1.1. Windows Phone 8

Importanța telefoanelor mobile a crescut în ultima perioada de timp foarte mult, iar acest segment a atras marii producători de tehnologie. Conform [1] cele mai multe dispozitive mobile rulează sistemul de operare Android, dezvoltat de Google. Pe următoarele locuri se situează iOS, Windows Phone ăi BlackBerry.

Windows Phone este un sistem de operare special dezvoltat de către Microsoft pentru telefoanele mobile. Ultima versiune a acestui sistem de operare este Windows Phone 8.1. Un telefon cu Windows Phone este un smartphone cu un sistem de operare Windows. Sistemul de operare Windows Phone 8 folosește același Windows NT kernel ca și Windows 8 [2].

Istoria Windows Phone începe în anul 2000 când Microsoft lansează Pocket PC 2000 și care are la bază sistemul de operare Windows CE 3.0. Acest sistem de operare suportă o rezoluție maximă de 240 x 320 pixeli și mai multe arhitecturi CPU. Făcând o paralelă între sistemul de operare Windows care există atunci, versiunea Pocket PC poate fi considerată Windows 98, Windows Millenium sau Windows 2000. Încă de la prima versiune a sistemului de operare pentru dispozitive mobile, Microsoft a inclus Pocket Office, Pocket Internet Explorer și Windows Media Player. Toate aceste programe suplimentare erau o versiune minimală a programelor ce rulau pe sisteme de operare desktop din acea perioadă.

În octombrie 2001 apare Pocket PC 2002, care va fi instalat pentru prima dată pe dispozitive smartphone. Această nouă versiune încearcă să fie asemănătoare ca

design cu Windows XP și aduce o serie de funcționalități noi precum suport pentru Virtual Private Network și MSN Messenger.

În 2003 apare Windows Mobile 2003 în 4 ediții, fiecare dedicată unui dispozitiv diferit. Windows Mobile 2003 pentru Pocket PC Phone Edition se adresează Poclet PCurilor care conțin funcționalități de telefon. În 2004 apare Windows Mobile 2003 Second Edition, în care apare ActiveSync (sincronizarea dispozitivului mobil cu un PC) și WPA. Un an mai târziu Microsoft lansează în cadrul unei conferințe Windows Mobile 5, cu un număr mare de funcționalități noi sau revizuite: Microsoft Exchange Server, un consum mai redus al bateriei dispozitivului mobil, interfață de management pentru GPS și suport pentru tastatură QWERTY.

După doi ani de dezvoltare a unui nou sistem de operare, Microsoft lansează cea mai stabilă versiune de sistem de operare de până atunci și anume Windows Mobile 6. Windows Mobile 6 apare în trei ediții: Windows Mobile 6 Standard (pentru telefoane mobile fără touchscreen), Windows Mobile 6 Professional pentru Pocket PC-uri cu funcționalitate de telefon și Windows Mobile 6 Classic pentru Pocket PC-uri clasice. Această ediție de Windows Mobile se bazează pe Windows CE 5.0, platforma strâns legată cu Windows Live și Exchange 2007. Dacă versiunile anterioare de Windows Mobile aveau un design asemănător cu Windows XP, Windows Mobile 6.0 are un design asemănător cu noul sistem de operare apărut în aceeasi perioadă și anume Windows Vista. Apar o serie de funcționalități noi precum VoIP, Windows Live, HTML email Outlook. Pe lângă aceste funcționalități, Microsoft preinstalează în memoria ROM a dispozitivelor mobile versiunea .Net Compact Framework si Microsoft SQL Server 2005 Compact Edition care sunt versiuni cu un set mai redus de funcționalități decât cele dezvoltate pentru desktop PC. Până la următoarea schimbare majoră a sistemului de operare urmează o serie de upgrade-uri: Windows Mobile 6.1, Windows Mobile 6.5 în 2009, Windows Mobile 6.5.1, Windows Mobile 6.5.3 și Windows Mobile 6.5.5 în 2010 însă doar pentru sporirea performanței (suport pentru noile formate de documente din Office 2007, eficientizarea ActiveSync, o serie de optimizări pentru IE) și reducerea consumului de baterie.

Următoarea versiune de sistem de operare pentru telefoane mobile aduce schimbări radicale fată de versiunile anterioare. Prima schimbare este aceea că Microsoft renunță la numele de Windows Mobile și numește noua versiune Windows Phone 7. Această versiune propune o interfață revizuită complet, include FM Radio și display multitouch cu o rezoluție de 800 x 480 pixeli. Datorită acestor schimbări din punct de vedere hardware, nu va fi posibil un upgrade de la versiunile anterioare de sistem de operare la Windows Phone 7. Chiar dacă Windows Phone 7 are un nume asemănător cu sistemul de operare pentru desktop PC-uri, Windows 7, acesta nu are la bază Windows 7 ci are la bază tot Windows CE ca și versiunile anterioare.

Noul sistem de operare încearcă să îmbine ideea de multitasking și multi-touch cu serviciile integrate Zune si Xbox Live [3]. O caracteristică importanta o reprezintă Windows Phone Hubs. Acestea sunt șase zone care înglobează aplicații diverse:

- People include aplicații pentru socializare
- Pictures include imagini pentru publicarea imaginilor și a clipurilor video pe rețelele de socializare
- Games conține jocurile Xbox Live
- Music/Video conține muzica, filme si radio FM
- Office suita de aplicați office dar cu o distribuție pentru telefoane
- Windows Store un magazin online din care se pot descărca sau cumpăra diverse aplicați

În figura 4.1 se poate observa interfața noua adusă de Windows Phone 7. Pe acest ecran se găsesc casete denumite Live Tiles aparținând aplicațiilor instalate pe dispozitivul mobil. Așa cum sugerează și numele, aceste casete se actualizează în timp real.



Figura 4.6 Ecranul de start pentru Windows Phone 7

Windows Phone 7 pune un accent destul de mare pe utilizarea rețelelor de socializare, integrând într-o singură aplicație, aplicația People, lista clasica de contacte cu "prietenii" de pe rețeaua de socializare Facebook. Tot din cadrul acestei aplicați se pot modifica setările contului de Facebook fără a fi nevoie să se intre pe site-ul oficial.

Microsoft preia o serie de facilități ale versiunii desktop de Outlook și creează Outlook Mobile. Această aplicație facilitează adăugarea conturilor de email de Windows Live, Google sau Yahoo, dar îmbunătățește și experiența de navigare printre mesaje.

Navigarea pe Internet se poate face cu versiunea mobile a Internet Explorer. Ce este interesant este că IE Mobile vine cu posibilitatea de a deschide simultan până la 6 taburi, însă această funcționalitate este prezentă și în alte browsere mobile precum Opera Mobile. Punctele slabe ale acestui browser sunt suportul pentru Flash și Silverlight.

Când a fost lansat Windows Phone 7 Android era deficitar la aplicațiile de tip office. Microsoft a dotat această versiune de Windows Phone cu Microsoft Office Mobile ce cuprinde versiuni mobile pentru Word, Excel și PowerPoint. Acestea se pot edita și salva pe Internet pentru a fi accesate ușor și pe versiunea desktop printr-o simplă descărcare sau chiar pe platforma online Office 365.

Un alt punct forte pe care mizează Microsoft este reprezentat de jocuri. Bazat pe succesul Xbox Live, Games Hub reprezintă punctul de plecare pentru jocurile disponibile pe Windows Phone 7, jucătorul având propriul profil și statistici pentru fiecare joc în parte pe care îl accesează.

Pe lângă toate aceste funcționalități noi aduse pentru utilizatori, Microsoft vine și în ajutorul dezvoltatorilor de aplicații. Pentru dezvoltarea aplicațiilor dezvoltatorii pot utiliza tehlologia Silverlight și XNA Framework pentru a construi aplicații cât mai interesante și mai complexe, profitând în același timp de caracteristicile fiecărui dispozitiv în parte. După crearea aplicațiilor dezvoltatorii au posibilitatea să comercializeze aplicațiile printr-un nou Windows Phone Marketplace care suportă plăti cu card, facturarea operatorilor mobili și aplicații finanțate prin publicitate.

Următoarea versiune de Windows Phone se numește Windows Phone 7.5 "Mango" care vine cu o versiune mobilă pentru Internet Explorer 9 care suportă aceleași standarde că și versiunea desktop. În PeopleHub se poate face acum și integrarea cu Twitter. Pe lângă aceste funcționalități noi se poate folosi și Windows Live SkyDrive.

În ianuarie 2013 apare un nou update pentru Windows Phone 7 și anume Windows Phone 7.8 care aduce facilități noi pentru ecranul de start incluse în versiunea Windows Phone 8 care a fost lansată între timp și mai multe teme de culori.

Înainte de a fi lansat update-ul Windows Phone 7.8 pentru Windows Phone 7, Microsoft lansează în iunie 2012 a treia generație a sistemului de operare Windows Phone și anume Windows Phone 8. Cea mai importantă schimbare adusă de nouă versiune este înlocuirea arhitecturii Windows CE cu arhitectura Windows NT care împarte multe componente cu Windows 8 pentru a ușura portarea aplicațiilor între cele două platforme. Odată cu schimbarea arhitecturii apare și imposibilitatea de a face update de la Windows Phone 7 la Windows Phone 8. Dacă Windows Phone 7 avea un singur tip de ecran, Windows Phone 8 permite mai multe tipuri de ecrane: 800x480, 1280x768 15:9 720p, 1280x720 16:9 1080p, 1920x1080 16:9. Alte noutăți pe care le aduce sunt suportul pentru procesoare multi-core, NFC, suport pentru spațiu de stocare adițional și o nouă îmbunătățire a ecranului de start conținând Live Tile-uri redimensionabile pe tot ecranul. Această versiune de Windows Phone a beneficiat de trei update-uri minore care au inclus fixare de defecte și îmbunătățiri mici precum radio FM sau conectivitate prin Bluetooth. Toate funcționalitățile prezentate dar și altele mai mici sunt descrise în detaliu în [4].

Următoare versiune de Windows Phone numită Windows Phone 8.1 se va apropia și mai mult de sistemul de operare Windows 8.1 și singurele modificări care vor trebui aduse unei aplicații de Windows 8.1 sau Windows Phone 8.1 vor fi doar cele ce vizează interfața utilizatorului, în rest fiind același cod sursă. O funcționalitate importantă este Cortana, un asistent personal virtual precum Google Now sau Siri de la Apple, dar care se integrează mult mai bine în sistemul de operare decât competitorii săi. În Windows Phone 8.1 va fi introdus un manager de fișiere nativ și o nouă versiune de Internet Explorer care va fi versiunea minimală la Internet Explorer 11. O funcționalitate foarte cerută este Action Center care există pe sistemele de operare competitoare de o perioada de timp semnificativă.

Aplicația Zonga este o aplicație mobilă ce rulează pe dispozitive instalate cu sistem de operare Windows Phone 8. Aplicația a fost dezvoltată în mediul de dezvoltare Visual Studio 2012, iar testarea s-a realizat cu ajutorul emulatorului instalat cu pachetul de dezvoltare și cu ajutorul a câtorva dispozitive fizice.

#### 4.3.1.2. Isolated Storage

Există momente când o aplicație stochează diferite date într-un fișier. Pentru acest fișier numele și locația în care va fi stocat trebuie alese cu grijă pentru a minimiza posibilitatea ca alte aplicații să cunoască locația lui și pentru a evita coruperea datelor. Fără un sistem standard care să gestioneze toate aceste probleme, dezvoltarea unor tehnici care să minimizeze conflictele de stocare poate fi o sarcină complexă, iar rezultatele să nu fie cele așteptate. Isolated Storage este un mecanism de stocare a datelor, care oferă, așa cum îi spune numele, izolare și siguranță, definind modalități standardizate de asociere a codului cu datele salvate.

Isolated Storage permite aplicațiilor să creeze și să mențină stocarea locală a datelor. Pentru o aplicație Windows Phone 8, toate operațiile de intrare/ieșire sunt

limitate la isolated storage și nu au acces direct la fișierele sistemului de operare sau ale altor aplicații instalate pe dispozitivul mobil. Datorită acestei funcționalități se îmbunătățește securitatea și reduce șansele de acces neautorizat la date și reduce șansa de corupere a lor.

Din [5] reiese că în funcție de tipul de date care trebuie stocate, aplicațiile Windows Phone 8 pot utiliza Isolated Storage în următoarele situații:

- Setări se pot stoca valori sub forma de cheie-valoare
- Fișiere și foldere
- Date relaționale stocarea datelor relaționale într-o baza de date SQL Server Compact Edition locală



Figura 4.7 Opțiuni de stocare a datelor cu Isolated Storage și clasele aferente

Suportul pentru Windows Phone este furnizat de către următoarele două spații de nume, ale căror caracteristici sunt prezentate in figura 4.2.

IsolatedStorageFile:

- System.IO.IsolatedStorage.IsolatedStorageFile ce permite crearea, folosirea și ștergerea fișierelor și directoarelor din isolated storage. Operațiile cu fișiere se realizează prin folosirea file stream-ului.
- System.IO.IsolatedStorage.IsolatedFileStream ce poate fi folosit pentru stocarea imaginilor, sunetelor și a fișierelor care sunt încărcate dinamic de pe Internet.
- •

Isolated Local Settings:

• System.IO.IsolatedStorage.IsolatedStorageSettings oferă un API pentru operații de citire/scriere și permite folosirea unor perechi de chei depozitate în isolated storage. Astfel, se creează o locație unică de stocare a datelor pentru fiecare utilizator al aplicației. Cu alte cuvinte, același dispozitiv poate avea mai multe locații de stocare pentru aceeași aplicație, presupunând ca are utilizatori diferiți. De asemenea, mai mulți utilizatori pot avea mai multe locații pentru isolated storage, cate una pentru fiecare aplicație.

Isolated Storage este o locație pe un dispozitiv cu Windows Phone 8 în care aplicația Zonga poate salva fișiere, informații de configurare, utilizatorul autentificat și alte tipuri de date la care orice altă aplicație nu are acces.

## 4.3.1.3. LINQ

Ultimele versiuni ale platformei Microsoft .NET și limbajele pe care această le suportă au ajuns la maturitate, însă mai rămân câteva probleme pentru dezvoltatori legate de accesarea datelor din surse diferite. Luând un caz particular, accesul la o bază de date și manipularea unui fișier XML sunt greoaie în cel mai bun caz și problematice în cel mai defavorabil caz.

În momentul în care se introduce o bază de date în cadrul unui sistem informatic această ridică numeroase probleme pentru dezvoltatori. În primul rând, nu se poate interacționa programatic cu o bază de date la nivelul nativ al bazei de date. Acest lucru înseamnă că erorile de sintaxă sau alte tipuri de erori nu pot fi detectate decât la rularea aplicație. Detectarea erorilor la rularea aplicație poate fi dezastruoasă și frustrantă în același timp deoarece se poate avea un întreg mecanism de tratare a excepțiilor iar aplicația să dea eroare din cauza unui cod invalid din punct de vedere sintactic. Uneori acest lucru este imposibil din cauza comportamentului neprevăzut al erorilor. Având cod care interacționează cu baza de date care nu este validat la momentul compilării poate duce cu siguranță la această problemă.

Microsoft a decis să facă un pas înainte și să rezolve aceste probleme ale dezvoltatorilor de sisteme informatice prin introducerea unui nivel de abstractizare suplimentar pentru interogarea datelor din aceste domenii de date particulare. Rezultatul a fost Language Integrated Query (LINQ). LINQ este o tehnologie care asigura un mecanism de interogarea a datelor de toate tipurile la nivelul limbajului de programare. Aceste tipuri includ șiruri și colecții de date din memorie, baze de date, documente XML și multe altele.

Chiar daca LINQ aduce multe avantaje pentru programatori, din [6] se poate observa că acesta introduce o latență suplimentară în citirea datelor care uneori poate fi importantă pentru unele sisteme informatice. Pentru îmbunătățirea performanței interogărilor se poate recurge la diferite tehnici precum cea descrisa in [7], însă nu este suficient pentru a egala performanta ADO.NET.

Conform [8] sunt 5 categorii de date in care poate fi utilizat LINQ:

- LINQ pentru obiecte
- LINQ pentru XML
- LINQ pentru seturi de date
- LINQ pentru SQL
- LINQ pentru entități

**LINQ pentru obiecte** este numele dat API-ului IEnumerable<T> pentru operatorii standard de interogare. Metodele statice din clasa System.Linq.Enumerable sunt cele care sunt folosite pentru a crea interogări pentru șiruri sau colecții de date care se află în memorie.

**LINQ pentru XML** este numele dat API-ului LINQ dedicat pentru lucrul cu XML-uri. Când Microsoft a adăugat librăriile pentru XML în LINQ aceasta a găsit și a fixat numeroase probleme în standardul XML DOM pentru a face că lucrul cu XMLuri să fie mult mai ușor. Odată cu apariția LINQ pentru XML nu mai este necesar să se facă parsarea documentelor cu ajutorul clasei XmlDocument. Pentru a beneficia de puterea LINQ pentru XML trebuie că într-un proiect să se importe System.Xml.Linq.dll și să se utilizeze directiva System.Xml.Linq.

**LINQ pentru seturi de date** este numele dat pentru API-ul LINQ care lucrează cu obiecte de tip DataSet. Sunt multe proiecte în care dezvoltatorii au utilizat obiecte de tip DataSet și pe care Microsoft nu dorește să îi excludă. Acești dezvoltatori nu vor trebui să rescrie funcționalități ale sistemelor informatice în care au folosit DataSet pentru a beneficia de puterea tehnologiei LINQ.

**LINQ pentru SQL** este numele dat API-ului IQueryable care oferă dezvoltatorilor posibilitatea să facă interogări pe baza de date Microsoft SQL Server. Utilizarea exclusivistă a serverului de bază de date de la Microsoft este un dezavantaj în cazul în care se dorește utilizarea unui alt server de baze de date precum MySQL. Pentru a utilizat LINQ pentru SQL trebuie să se facă o referință către System.Dată.Linq.dll și să se utilizeze directiva System.Dată.Linq.

LINQ pentru entități este un API LINQ alternativ pentru interogarea unei baze de date. Decuplează modelul entității de modelul fizic din baza de date prin injectarea unor mapări logice între cele două modele. Odată cu această decuplare crește puterea, flexibilitatea și complexitatea interogărilor care se pot efectua asupra bazei de date. LINQ pentru entități este recomandat să se utilizeze în cazul în care este nevoie de mai multă flexibilitate sau mai multă decuplare decât permite LINQ pentru SQL.

Tehnologia LINQ a fost folosită în cadrul aplicației Zonga pentru a obține o abstractizare asupra modelului fizic de date și pentru crearea unor interogări mai complexe într-o perioada de timp mai mică.

## 4.3.1.4. Servicii Web

Termenul de "serviciu web" are un înțeles imprecis supus mereu schimbărilor și redefinirilor. După cum sugerează numele "serviciu web" este o aplicație disponibila pe Internet, adică o aplicație care rulează in mod tipic utilizând protocolul HTTP [9]. Conform World Wide Web Consortium, "un serviciu web este un sistem software conceput pentru a facilita comunicarea între mașini distincte prin intermediul unei rețele". Alternativ, Zeldman definește un serviciu web ca "un set de componente software reutilizabile bazate pe XML și alte protocoale asemănătoare care permit interacțiunea de-a lungul unui ecosistem de afaceri cu un cost aproape zero" [10, p. 112].

Câteva trăsături sunt totuși specifice serviciilor web. Dintre cele prezentate în [11] se pot enumera următoarele:

- **Infrastructura deschisă**. Serviciile web sunt folosite cu ajutorul unor protocoale definite de standarde specifice industriei, independente de dezvoltator, precum HTTP și XML. Acestea se pot folosi de protocoale existente, formate de date și politici de securitate. Această caracteristică uşurează folosirea lor și promovează interoperabilitatea între sisteme.
- **Transparența limbajului**. Serviciile web și clienții lor pot comunica indiferent de limbajul de programare în care au fost descrise serviciile respective. Limbaje precum C#, Java, C, Python, JavaScript și altele oferă biblioteci și chiar framework-uri care facilitează interacțiunea cu serviciile web.
- **Design modular**. Serviciile web sunt modulare în design astfel încât servicii noi pot fi generate prin integrarea si suprapunerea serviciilor existente deja.

Serviciile web sunt inerent sisteme distribuite. Schimbul de informație între producătorul mesajului și destinatarul său se poate realiza cu ușurință dacă cele două componente știu să codifice/ decodifice mesajul într-un anumit standard, de exemplu

XML. Altfel, schimbul de informație s-ar realiza cu dificultate. Chiar dacă în majoritatea cazurilor schimbul de informație se face sub formă de text, trebuie amintit faptul că se poate transmite și conținut în format binar dacă este necesar.

Serviciile web pot fi împărțite în două categorii principale: REST (REpresentational State Transfer) și SOAP (Simple Object Access Protocol).

În continuare va fi descrisă doar categoria de servicii REST deoarece aceasta a fost folosită în cadrul aplicației Zonga pentru Windows Phone 8.

Primul aspect de notat în ceea ce privește serviciile REST este că acest tip de servicii își propun să expună date codificate sub formă de text spre deosebire de SOAP care folosește standardul WDL pentru a expune o interfață. Rest se folosește de protocolul HTTP pentru a transmite date.

După cum rezultă din denumirea lor, aceste servicii expun resurse care au o stare. Resursele pot fi stocate pe un calculator, într-o bază de date ca un flux de biți, un document, o înregistrare într-o bază de date etc. Orice resursă are cel puțin un URI (Uniform Resource Identifier) prin intermediul căruia poate fi accesată pe Internet în mod unic [12].

Un client HTTP manipulează o resursă prin faptul că se conectează la serverul care o deține și trimite o cerere de tip GET împreună cu o cale către resursă. De clientul doreste obtină textul exemplu. dacă să aflat la locatia www.exemplu.ro/salut.txt, atunci el trebuie să formeze o cerere la locatia www.exemplu.ro și să ceară resurse aflate la locația /salut.txt. Presupunând că resursa contine textul "Salut!", atunci următorul tabel ilustrează cererea clientului și răspunsul serverului:

Cerere client	Raspuns server
GET /salut.txt HTTP/1.1	200 OK
Host: www.exemplu.ro	Content-Type: text/plain
	Salut!

Figura 4.8 Exemplu de cerere și răspuns HTTP

Acest tip de servicii se bucură de faptul că au o interfața unică, dată de protocolul HTTP:

- Obținerea de informații se face prin metoda HTTP GET
- Crearea unei noi resurse se face prin HTTP PUT pentru un URI nou, sau HTTP POST pentru un URI existent
- Modificarea unei resurse existente se face cu ajutorul metodei HTTP PUT.
- Ștergerea unei resurse se face cu ajutorul metodei HTTP DELETE

Verb HTTP/URI	Semnificatia CRUD
POST mus	Creaza o noua melodie
GET mel	Citeste toate melodiile
GET mel?id=abc	Citeste melodia abc
PUT mel	Face un update asupra melodiei
DELETE mel	Sterge toate melodiile
DELETE mel?id=abc	Sterge melodia abc

Figura 4.9 Folosirea verbelor HTTP pentru accesarea resurselor pe server

Trebuie să amintim de siguranța și idempotența resurselor. Siguranța se referă la faptul că prin accesul unei resurse, starea ei nu a fost modificată astfel încât să devină invalidă. Idempotența se referă la faptul că o resursă poate fi accesată ori de câte ori iar rezultatul va fi același. GET, HEAD, PUȚ și DELETE sunt idempotențe. Aceste două trăsături permit unui client să efectueze cereri sigure pe HTTP. DE exemplu, în cazul în care clientul nu a primit un răspuns la o cerere GET, atunci o mai poate efectua o dată – ea este atât sigură cât și idempotență!

Serviciile REST pot expune date în diferite maniere: text, XML, JSON sau HTML. Cea mai comună metodă de expunere a datelor este reprezentarea sun formă de XML. XML (Excensible Markup Language) reprezintă un sistem de marcare similar cu HTML, doar că este mult mai cuprinzător și mai dinamic, diferența esențială fiind că etichetele nu sunt definite, dezvoltatorii fiind liberi să își creeze propriile structuri.

JSON (JavaScript Object Notation) este un format de reprezentarea și interschimbare a datelor. Este în format text, inteligibil pentru oameni și este utilizat pentru reprezentarea obiectelor și a altor structuri de date. Este folosit în special pentru a transmite date structurate prin rețea printr-un proces de serializare. JSON este o alternativă mai simplă și mai facilă a XML-ului în procesul de transmitere a datelor. Eleganța formatului JSON provine din faptul că este un subset al limbajului JavaScript.

În figura 4.3 este ilustrată reprezentarea unei melodii în cele două forme descrise mai sus.

```
XML:

<song>

<id>aby6rfj6</id>

<name>Song1</name>

<duration>300</duration>

</song>

JSON:

{

"id":"aby6rfj6",

"name":"Song1",

"duration":300

}

Figura 4.10 XML vs JSON
```

Ușurința în accesarea datelor expuse, precum și modul lor comun de reprezentare recomandă serviciile web pentru comunicarea cu clienții de diferite tipuri și arhitecturi. Luând în considerare acest aspect, putem concluziona că aplicațiile dezvoltate pentru platforma Windows Phone 8 pot consuma cu succes serviciile web de tip REST. Manierea de comunicare între aplicația Zonga pentru Windows Phone 8 și sistemul Zonga se va face cu ajutorul standardului JSON.

#### 4.3.1.5. JSON.NET

JSON.NET este cel mai popular framework adițional pentru framework-ul Microsoft .NET. JSON.NET serializează/deserializează orice obiect .NET într-un obiect JSON cu ajutorul serializatorului său. Acest framework, pe lângă funcționalitatea să de baza, oferă suport și pentru LINQ pentru standardul JSON cu ajutorul claselor JObject, JArray și JValue. Cu ajutorul JSON.NET putem să definim o schemă JSON și să o aplicăm pe răspunsul primit de la server pentru validarea acestuia înainte de a fi parsat. Pe lângă aceste funcționalități mai este posibilă conversia între standardul JSON și XML în ambele sensuri.

## 4.3.2. Design pattern-uri

#### 4.3.2.1. MVVM

Odată cu începerea dezvoltării de interfețe utilizator s-au remarcat multe design pattern-uri celebre pentru a le ușura munca. De exemplu pattern-ul Model-View-Presenter (MVP) este un pattern foarte folosit în multe limbaje de programare pentru interfețe utilizator. MVP este o variație a pattern-ului Model-View-Controller care este utilizat de câteva decenii.

În 2005, John Grossman, unul dintre arhitecții Windows Presentation Foundation (WPF) și Silverlight publica pe blogul sau pattern-ul Model-View-ViewModel (MVVM) [13]. MVVM este identic cu MVP și ambele pattern-uri oferă o abstractizare a unei interfețe utilizator, ce conține starea și comportamentul acesteia. MVVM este o specializare a pattern-ului Presentation Model (PM), creat pentru a se integra mai bine în cadrul platformelor WPF și Silverlight. Această specializare se poate observa și din articolul [13] în care Glenn Block nu folosește niciodată termenul de ViewModel. În schimb, folosește termenul Presentation Model pentru a descrie o abstractizare a unei interfețe utilizator.

Spre deosebire de componenta Presenter a pattern-ului MVP, un ViewModel nu are nevoie de o referință la interfața utilizator. Interfața utilizator se leagă la ViewModel prin mecanismul de *binding* și care expune obiecte de tip model și stări specifice interfeței utilizator. Legătura dintre ViewModel și model este simplă de construit deoarece obiectul de tip ViewModel este setat că și obiect de context pentru interfața utilizator. Dacă valorile proprietăților din ViewModel se modifică, noile valori sunt propagate automat către interfața utilizator prin intermediul mecanismului legătură de date. Când un utilizator apasă un buton pe interfața utilizator în ViewModel se execută o comandă pentru a execută o acțiune specifică. Instanța de ViewModel face modificări asupra obiectelor de tip model și niciodată interfața utilizator. În figura 4.4 sunt reprezentate clasele descrise mai sus care compun pattern-ul MVVM și modul de interacțiune între ele.



Figura 4.11 Clasele MVVM si interactiunea dintre ele

Obiectele de tip interfață utilizator nu știu nimic de clasele de model, în timp ce obiectele de tip ViewModel și model nu știu nimic de existența unei interfețe utilizator.

De fapt obiectele de tip model nu știu nici de obiecte ViewModel și nici de interfețele utilizator în care sunt folosite. Acesta este o parte a arhitecturii cu legături foarte slabe între componentele aplicației.

## **4.3.2.2.** Abstract Factory

Pattern-ul Abstract Facory oferă o modalitate de a îngloba un set de metode de fabricare care au ceva în comun fără a specifica clasele lor concrete. În condiții obișnuite de utilizare, software-ul creează o implementare a clasei abstracte și apoi utilizează interfața generică de fabricare pentru a crea clase concrete care fac parte din același set. Clientul nu știe care obiecte concrete le folosește din moment ce el folosește numai interfețe generice produse de acesta. Acest pattern separă detaliile de implementare a unui set de obiecte de uzul lor general și se bazează pe compoziția obiectelor.



Figura 4.12 Diagrama UML a pattern-ului Abstract Factory

În figura 4.7 este reprezentată diagrama UML a pattern-ului și în a cărui componență intră:

- AbstractFactory: declară o interfață pentru operațiile de creare a obiectelor concrete
- **ConcreteFactory**: implementează operațiile de creare a obiectelor concrete
- AbstractProduct: declară o interfață pentru un tip de obiect concret
- **ConcreteProduct**: definește un obiect concret creat de către metoda de fabricare corespunzătoare
- **Client**: utilizează doar interfețele declarate de către clasele AbstractFactory și AbstractProduct

## Capitolul 5. Proiectare de Detaliu și Implementare

În acest capitol va fi descris în detaliu modul de implementare al aplicației. Se va detalia modul de proiectare și implementare a elementelor de bază din cadrul aplicației, cum ar fi: realizarea arhitecturii conceptuale, a diagramelor de secvență, identificarea și descrierea componentelor, interacțiunea dintre aplicație și sistemul de operare și a proceselor implicate în comunicarea cu serverele sistemului.

#### 5.1. Arhitectura sistemului Zonga

Sistemul Zonga este implementat în conformitate cu arhitectura client-server. Arhitectura client-server reprezintă o rețea de calculatoare în cadrul căreia mai mulți clienți solicită și primesc servicii de la un cluster centralizat de servere. Acest tip de arhitectură este cel mai frecvent model de arhitectură folosit de către dezvoltatori, devenind, astfel, cel mai important model pentru crearea de aplicații distribuite.

În cadrul acestui model sesiunea de comunicare este deschisă de către un client, prin intermediul unei cereri către cluster-ul de servere. Unul dintre serverele din acel cluster primește cererea, realizează operațiile solicitate de către clienți și le trimite acestora un răspuns.

La modul ideal, server-ele ar trebui să dispună de o interfață transparentă pentru clienți, astfel încât aceștia să nu aibă cunoștință de specificațiile software și hardware ale server-elor care furnizează serviciile. În cadrul sistemului Zonga această interfață transparentă este realizată prin intermediul unui API (Application Programming Interface) JSON cu ajutorul căruia clienții comunică cu serverele centrale.

De obicei, clienții sunt situați în stații de lucru sau pe computerele personale sau în buzunarele utilizatorilor, în timp ce serverele sunt situate în altă parte în rețea, de obicei, pe mașini mult mai puternice, iar informațiile serverului pot fi accesate de mai mulți clienți concomitent.



Figura 5.1 Arhitectura Client-Server a sistemului Zonga

Acest model de calcul este deosebit de eficient atunci când clienții și serverul au sarcini distincte. De exemplu, în cadrul sistemului Zonga, un client afișează informații referitoare la topurile din cadrul sistemului, la recomandările unui utilizator și redarea fișierelor audio, în timp ce pe servere se execută algoritmi pentru crearea de recomandări, algoritmi pentru generarea topurilor, dar și gestionează baza de date în care sunt stocate toate informațiile persistente.

Structura de ansamblu a sistemului Zonga se poate observa cel mai bine prin intermediul unei diagrame (figura 5.1). După cum se poate observa în cadrul figurii, există două componente distincte: serverele și clienții, care pot fi laptop-uri sau diferite tipuri de dispozitive mobile cu diferite sisteme de operare. Serverul central comunică cu clienții, în timp ce clienții nu comunica unul cu altul. Comunicarea între clienți se face prin intermediul server-ului sistemului.

## 5.1.1. Rolurile componentelor sistemului

Pentru a clarifica ce face fiecare componentă a sistemului, se va prezenta imaginea de ansamblu cu privire la distribuirea celor mai importante funcționalități între componente.

S-a încercat să se realizeze o decuplare cât mai mare între componentele sistemului și să se execute cât mai multe operații pe partea de server. Această execuție a celor mai importante operații pe partea de server are mai multe avantaje precum: modificarea aplicației server se va face mult mai ușor decât modificarea fiecărui client în parte, o arhitectură centralizată este mai puțin complexă decât o arhitectură distribuită și nu în ultimul rând dispozitivele mobile au o putere de procesare mai mică atingându-și limitele foarte repede.

## 5.1.1.1. Aplicațiile client

Așa cum s-a menționat anterior, majoritatea operațiilor se fac pe partea de server. Astfel, clienții sunt responsabili mai mult sau mai puțin de afișarea conținutului primit de la server și trimiterea acțiunilor utilizatorului la serverul sistemului. Următoarea listă conține câteva detalii despre operațiile pe care trebuie să le facă o aplicație client.

- Trimiterea de informații la server cu privire la acțiunile utilizatorilor; acțiuni precum ascultarea unei melodii, trecerea la o alta melodie, ascultarea unei melodii în mod offline, crearea unui playlist, adăugarea sau ștergerea din categoria favorite a unui element etc.
- Afișarea conținutului primit de la server. Aceasta include informații legate de recomandările unui utilizator, topurile sistemului, elementele favorite ale utilizatorului etc.
- Descărcarea melodiilor de pe server și salvarea lor în suportul de stocare al dispozitivului
- Redarea melodiilor fie din suportul de stocare fie de direct de pe Internet

## 5.1.1.2. Aplicația server

Următoarea listă conține operațiile care trebuie efectuate de către aplicația server.

- Gestionarea utilizatorilor sistemului
- Trimiterea de informații la clienți pe baza cererilor create de aceștia

- Gestionarea topurilor
- Gestionarea elementelor recomandate pentru fiecare utilizator
- Gestionarea și persistarea informațiilor într-o bază de date

## 5.2. Arhitectura aplicației server

În acest subcapitol se va prezenta arhitectura conceptuală a aplicației server care rulează în centrul de date Zonga. Chiar dacă aplicația server nu se regăsește printre obiectivele acestei lucrări, este necesară o descriere a sa deoarece aplicația client va folosi o serie de funcționalități incluse în cadrul aplicației server și pentru a oferi o imagine de ansamblu asupra împărțirii funcționalităților pe componentele sistemului.



Figura 5.2 Arhitectura aplicație Zonga server

În figura 5.2 este ilustrată arhitectura aplicației Zonga server la nivel de componente, iar în continuare va fi descrisă pe scurt fiecare componentă în parte.

**Componenta principală** preia toate cererile venite de la clienți, verifică dacă poate formula un răspuns pentru cererea clientului, dacă nu poate răspunde interoghează altă componentă responsabilă pentru acea funcționalitate. În momentul În care aplicația are un rezultat la cererea clientului, formulează un răspuns și îl trimite acestuia. Funcționalitățile pe care le deservește componenta principală sunt gestionarea utilizatorilor și a setărilor acestora, gestionarea playlist-urilor unui utilizator, gestionarea favoritelor și este responsabilă pentru trimiterea melodiilor către clienții mobili.

**Componenta pentru gestionarea topurilor** generează la un anumit interval de timp lista cu cele mai ascultate melodii, cu cele mai ascultate albume și cu cei mai ascultați

artiști de-a lungul timpului, pe baza unor statistici privind numărul de ascultări a utilizatorilor platformei.

**Compienta de recomandare** al sistemului Zonga este una proprie și generează recomandări în timp real pentru fiecare utilizator. Pentru recomandările de melodii, albume, artiști și playlist-uri, componenta de recomandare folosește parametri precum genul de muzică ascultat, numărul de ascultări ale unui artist, câte melodii s-au ascultat dintr-un anumit gen, ce tip de artist a ascultat utilizatorul, ce melodii favorite are un utilizator s.a.m.d.

Persistarea datelor din cadrul sistemului se face într-un cluster de baze de date MongoDB.

#### 5.3. Aplicația Zonga pentru Windows Phone 8

Arhitectura aplicației Zonga pentru Windows Phone este prezentată în figura 5.3. Modelul arhitectural pe care este realizată aplicația este layer. Aplicația este compusă din cinci componente grupate pe trei layere și suportul de stocare. Utilitarul pentru request-uri transformă obiectele de tip model în JSON și face cererea către server iar la primirea răspunsului verifică dacă cererea s-a efectuat cu succes și transformă răspunsul JSON primit în obiecte de tip model. Componenta de servicii efectuează operații mai complexe cu ajutorul modelelor și utilizează suportul de stocare pentru salvarea anumitor informații. Componenta de conversie, transformă obiectele de tip model în obiecte de model pentru interfața utilizator. Nivelul de prezentare afișează rezultatele utilizatorului în funcție de acțiunile pe care acesta le efectuează. Cea mai importantă componentă a aplicației este componenta pentru redarea melodiilor. Aceasta a fost plasată pe diagramă arhitecturală între sistemul de operare și aplicație datorită faptului că sistemul de operare gestionează acțiunile utilizatorului iar aplicația este notificată cu privire la acestea pentru a lua anumite decizii în redarea melodiilor.



Figura 5.3 Arhitectura aplicație Zonga pentru Windows Phone 8

În continuare vor fi descrise toate componentele aplicație și modul în care acestea comunică între ele.

#### 5.3.1. Utilitarul pentru request-uri

Aproape toate funcționalitățile aplicației au nevoie de conexiune la Internet pentru a funcționa. Singura excepție este cazul în care un utilizator ascultă melodiile descărcate pe suportul de stocare al telefonului său. Toate cererile la server se efectuează într-un thread separat pentru a nu bloca interfața utilizatorului. Thread-ul așteaptă până când apare mesajul de răspuns de la server și apoi trimite mesajul înapoi la obiectul de tip serviciu care a dorit o informație de la server.

Pentru a efectua o cerere la server, componenta creează un obiect de cerere și setează anumite câmpuri precum: numele metodei apelate de pe server și parametri acesteia. Următorul pas este serializarea obiectului în format JSON și trimiterea cererii la server. În urma cererii serverul generează un răspuns JSON care fie conține informațiile necesare, fie conține un mesaj de eroare pentru utilizator. Răspunsul se deserializează sub forma unui obiect de răspuns și se verifică dacă conține o eroare. Dacă se detectează eroare se va crea o excepție specifică în funcție de mesajul de eroare primit și se va afișa utilizatorului. În cazul în care răspunsul nu conține o eroare, acesta se deserializează într-un obiect de model și se va returna obiectul serviciului care a dori informația respectivă de la server.

#### 5.3.2. Nivelul de servicii

Nivelul se servicii este responsabil de procesarea suplimentară care trebuie efectuată asupra datelor primite de la server . În cadrul acestui nivel, fiecare cerință funcțională are corespondent o clasă de serviciu. Luând ca exemplu cerința funcțională ce se referă la gestionarea playlist-urilor unui utilizator avem clasa PlaylistService. Această clasă este responsabilă pentru crearea unui playlist, pentru ștergerea lui, pentru redenumirea lui și pentru descărcarea sa.

Toate serviciile aplicației interacționează cu componenta utilitară pentru crearea cererilor către server cu ajutorul căreia se obțin un set de date sub forma unor obiecte de tip model. În funcție de obiectele obținute ca răspuns, fiecare serviciu generează un alt obiect de tip model specific unei pagini din cadrul interfeței utilizator care conține doar câmpurile care urmează să fie afișate.

#### 5.3.3. Nivelul de prezentare

Scopul acestei secțiuni nu este să se prezinte interfața grafică ci să se motiveze alegerile de design făcute și cum au fost construite diferite elemente și mecanisme pentru crearea unei experiențe cât mai bună în cadrul aplicației.

Pentru fiecare cerință funcțională s-a creat o pagină separată în cadrul căreia vor fi afișate informații utilizatorului. Prin intermediul paginilor, utilizatorii aplicației vor putea efectua anumite operații în funcție de pagina pe care se află.

Fiecare pagină a fost dezvoltată conform pattern-ului MVVM care a fost descris în capitolul 4.3.2.1. Acest pattern asigură o dezvoltare cât mai decuplată între logică aplicației și interfața utilizator. Așa cum a fost precizat mai devreme pentru fiecare secțiune a unei

pagini se utilizează un model simplificat care conține doar informațiile care se vor afișa utilizatorului.

Una dintre cerințele non-funcționale a fost să se optimizeze traficul efectuat către serverele sistemului. Această constrângere a fost realizată încărcând fiecare secțiune din cadrul unei pagini cât mai târziu posibil. Prin această încărcare întârziată dacă utilizatorul nu dorește să vizualizeze o anumită secțiuni atunci nu se va efectua cererea la server.

Majoritatea secțiunilor unei pagini conțin o listă în care se afișează melodii, albume, artiști sau playlist-uri. Aceste liste pot să conțină câteva mii de elemente care trebuie afișate unui utilizator . Un număr așa mare de elemente poate epuiza foarte repede memoria unui telefon și trebuia găsit un mecanism care să nu încarce în memorie decât elementele care vor fi afișate utilizatorului cu o marjă de câteva elemente în plus care să poate fi afișate până se încarcă altele. Această constrângere a fost realizată implementând o colecție specială numită "DataVirtualizationCollection" care realizează cerința descrisă mai sus.

Colecția "DataVirtualziationCollection" oferă dezvoltatorului un set de evenimente care se vor lansa în anumite momente ale utilizării colecției pentru că dezvoltatorul să poată efectua un set de operații precum adăugarea unor noi elemente în colecție când aceasta nu mai are elemente de afișat. Pentru a înțelege mai bine cum funcționează se va descrie un scenariu de afișare a melodiilor din cadrul topurilor sistemului. La încărcarea paginii, și implicit a listei care utilizează colecția, obiectul de tip colecție lansează evenimentul de "ItemsRealized". La lansarea acestui eveniment, dezvoltatorul este semnalat că este nevoie de mai multe elemente pentru afișare. Dacă se încarcă elemente atunci acestea vor fi afișate utilizatorului, altfel acest eveniment nu se va mai lansa pentru această secțiune decât la o nouă încărcare a paginii. În momentul în care utilizatorul face un gest pe ecran și se depășește numărul de elemente afișate utilizatorului, colecția va încărca în memorie un anumit număr de elemente din cele salvate pe disk pentru a fi afișate , iar elementele care au ieșit din zona vizuală a utilizatorului vor fi salvate pe disk. Dacă se ajunge la sfârșitul colecției se va declanșa un eveniment de "ItemsRealized" în care dezvoltatorul are opțiunea de a adăuga alte elemente în colecție sau nu.

Stilul în care este realizată interfața utilizatorului este unul intuitiv, simplu și care respectă principiile impuse de către Microsoft în ceea ce privește design-ul aplicațiilor Windows Phone 8. Stilul de culoare este unul simplu și folosește aceeași paletă de culori în cadrul întregii aplicații. Culoarea de fundal de pe prima pagină este neagră cu o poză pe fundal, iar pe restul paginilor culoarea este gri. Textul are diferite nuanțe de gri și albastru, iar culoarea de fundal a elementelor este albă. Aceste culori au fost alese intenționat deoarece oferă un contrast bun între elementele unei pagini și este ușor de citit atât pe întuneric cât și în lumina puternică a soarelui. Albastru a fost folosit deoarece este culoarea logo-ului. Un alt motiv pentru care s-a ales această paleta de culori este pentru reducerea consumului de energie al aplicației.

În ultimul rând, se vor prezenta câteva aspecte despre tratarea erorilor. În cazul general în momentul în care apare o eroare pe partea de server sau în cadrul aplicației utilizatorul este notificat prin intermediul unei notificări despre eroarea care a apărut dacă această poate fi identificată. Dacă o eroare nu a putut fi identificată se afișează un mesaj de eroare generic "A apărut o eroare. Va rugăm reîncercați". Este posibil că în unele cazuri anumite erori să nu poată fi tratate de către aplicație și aceasta să se închidă. În acest caz a fost integrat în cadrul aplicației sistemul BugSense care trimite la un server al sistemului

BugSense cauza erorii, stacktrace-ul excepției și anumite detalii despre dispozitivul utilizatorului.

## 5.3.4. Utilitarul pentru baza de date

Aplicația folosește o baza de date pentru a stoca melodiile care au fost descărcate de către un utilizator, numele playlist-ului din care fac parte și numele utilizatorului care a descărcat melodiile respective. Chiar dacă baza de date este una care conține doar o singură tabela, a fost necesară dezvoltarea acestei componente separate pentru a separa accesului la baza de date de restul aplicației.

Toate interogările sunt scrise în această componentă a aplicației client. Baza de date este una de tip relațional. Având în vedere că în cadrul fiecărei secțiuni se utilizează doar un serviciu care poate folosi baza de date, nu a fost nevoie de utilizarea unei librarii pentru gestionarea conexiunilor și paralelizarea accesului.

## 5.3.5. Componenta pentru redarea melodiilor

Cea mai complexă componentă a acestei aplicații este componenta pentru redarea melodiilor. Pentru redarea continuă a melodiilor în orice pagină a fost nevoie de utilizarea player-ului din background a sistemului de operare. La acest player managementul îl face sistemul de operare iar dezvoltatorii au la dispoziție un set de clase pe care le pot extinde și a căror metode sunt apelate la anumite acțiuni ale utilizatorului pentru a se lua decizii cu privire la redarea melodiilor. Din această cauză componenta pentru redarea melodiilor a fost plasată între sistemul de operare și aplicație. Având în vedere că aplicația suportă două tipuri de redare a melodiilor: redare a melodiilor salvate pe disk și a melodiilor direct de pe Internet se disting două cazuri care vor fi prezentate mai jos.

În cadrul etapei de dezvoltare s-a ținut cont de faptul că, componenta de redare a melodiilor trebuie să nu depășească 15 Mb de memorie pe care o utilizează. Această condiție este impusă de către sistemul de operare, iar în cazul în care nu se respectă nu va mai apela nici o metodă din cadrul claselor care sunt extinse pentru a reda o melodie.



## 5.3.5.1. Redarea melodiilor salvate local

Figura 5.4 Componentele utilizate pentru redarea melodiilor salvate local

În figura 5.4 sunt reprezentate toate componentele care intră în componența redării melodiilor salvate pe dispozitivul mobil al utilizatorului. Redarea unei melodii funcționează în felul următor: un utilizator dorește să asculte o melodie sau un set de melodii salvate local. În momentul în care el apasă butonul de play sau dă tap pe o melodie din oricare pagină a aplicației, se va salva în suportul de stocare lista de melodii care urmează să fie redată. După această acțiune se notifică sistemul de operare că aplicația e gata pentru redarea unei melodii. Sistemul de operare va apela o serie de metode din cadrul componentei "Player de melodii" care va încarca lista de melodii din suportul de stocare și va decide care melodie urmează să fie redată pe baza setărilor utilizatorului.



Figura 5.5 Schema logică simplificată pentru redarea melodiilor

#### Capitolul 5

Componenta "Player de melodii" va specifica sistemului de operare ce melodie dorește să fie redată și că poate începe redarea unei melodii. În continuare sistemul de operare va notifica componenta "MediaSource" despre melodia care va trebui redată. Această componentă fiind utilizată atâta pentru redarea melodiilor de pe Internet cât și a celor salvate local va decide ce componentă va utiliza: pentru redarea melodiilor de pe Internet sau a celor salvate local. În cazul de față va decide utilizarea componentei pentru redarea melodiilor salvate local. Pasul următor este notificarea componentei "Buffer Manager" despre melodia care trebuie încărcată în buffer și punerea sub supraveghere a gradului de încărcare a memoriei tampon. Buffer Manager-ul vede că buffer-ul este gol și notifică cititorul de date să citească următoare secventă de biți pentru a fi încărcată în buffer. Cititorul de date citește datele de pe disk, le decriptează și le pune în memoria tampon. În timp ce componenta de citire a datelor încarcă în memorie datele, componenta "MediaSource" tot cere parser-ului de melodii MP3 - melodiile salvate local sunt doar MP3 așa cum e descris în secțiunea 5.4.1 – următorul frame de MP3 care va fi trimis înapoi la sistemul de operare pentru a fi redat. În cazul în care parser-ul nu îi poate oferi componentei "MediaSource" un frame într-o anumită perioada de timp, acestuia nu îi va mai fi cerut un alt frame si se va abandona redarea melodiei respective.

Componenta Buffer Manager în timpul redării unei melodii este responsabilă de gradul de încărcare a memoriei tampon și atunci când gradul de ocupare al memoriei scade sub 35% trebuie să încarce următorul segment al melodiei în memoria tampon dacă mai există unul.

La terminarea melodiei MediaSource notifică sistemul de operare care la rândul său notifică componentă "Player de melodii" și ciclul se repetă.

Algoritmul descris mai sus este ilustrat prin intermediul unei scheme logice reprezentate în figura 5.5.

#### 5.3.5.2. Redarea melodiilor de pe Internet

Având în vedere că redarea melodiilor de pe Internet este destul de asemănătoare cu redarea melodiilor de pe suportul de stocare al dispozitivului mobil se vor descrie doar diferențele dintre aceste două tipuri de redări. Așa cum se poate observa în figura 5.5 cititorul de date va prelua datele de pe Internet cu ajutorului unei componente utilizate pentru a efectua cereri la server.

Componenta de decriptare va fi folosită doar în cazul melodiilor M3U8, în rest nefiind necesară deoarece doar melodiile M3U8 sunt criptate.

În locul parser-ului de MP3 a apărut o componentă generică Parser care poate fi înlocuită de oricare dintre cele trei parsere: M3U8 parser, MP3 parser şi AAC parser. De data aceasta componenta MediaSource va crea o instanță de parser cu ajutorul unei componente "Parser Factory" responsabilă pentru crearea parserului corespunzător în funcție de extensia fișierului care conține melodia ce trebuie redată. Nu s-a implementat o detecție a parser-ului independentă de extensia fișierului deoarece introducea o complexitate suplimentară și afecta performanța aplicației.



Figura 5.6 Componentele utilizate pentru redarea melodiilor de pe Internet

## 5.4. Detalii de implementare

#### 5.4.1. Descărcarea playlist-urilor

Una dintre cele mai importante constrângeri în redarea melodiilor este încadrarea în 15 MB de memorie consumată în timpul redării unei melodii. Această constrângere impune anumite reguli și asupra modului în care vor fi salvate piesele în urma unei descărcări. Toate melodiile descărcate vor fi de tip MP3. Având în vedere că majoritatea melodiilor ocupă în jur de 7MB se poate observa că dacă s-ar încerca salvarea întregii melodii, la redarea sa va fi nevoie de un buffer intermediar de 7 MB pentru încărcarea sa în memorie și încă 7 MB pentru un buffer intermediar care va tine melodia decriptată, se ajunge rapid la limita de memorie impusă.

O soluție la această problemă este împărțirea fișierului descărcat în bucăți mai mici astfel încât suma buffer-elor necesare pentru redarea unei melodii să nu depășească limita de memorie impusă în cadrul componentei de redare. S-a decis împărțirea melodiilor în bucăți de câte 1 MB fiecare. Înainte de salvarea pe suportul de stocare a unui segment acesta trebuie criptat datorită constrângerilor impuse de legea drepturilor de autor asupra melodiilor pe care le au casele de discuri.

Un scenariu pentru salvarea unui playlist este descris prin intermediul diagramei de secvență ilustrată în figura 5.6.



Figura 5.7 Diagrama de secvență pentru descărcarea unui playlist

## Capitolul 6. Testare și Validare

Testarea s-a realizat în mod incremental după fiecare etapă de proiectare, după implementarea fiecărei componente sau funcționalități în parte. Maniera în care s-a realizat testarea este de la scenarii simple de test la scenarii complexe test.

Cunoscând toate cerințele aplicației, s-a testat ca fiecare funcționalitate să fie operațională și s-au căutat erori în fiecare scenariu. Pentru verificarea atingerii tuturor obiectivelor aplicației și a corectitudini lor, s-a făcut testare la nivel de interfață, prin selectarea de scenarii de utilizare.

Testarea aplicației s-a realizat în mod manual pe baza scenariilor de test create în conformitate cu funcționalitățile aplicației. Testare automată s-a realizat doar în cadrul componentei pentru redarea melodiilor și s-a efectuat cu ajutorul uneltei Microsoft Test Framework.

În continuare se vor descrie doua seturi de scenarii de test care au fost utilizate pentru testarea aplicației.

#### TC1. Titlu: Creare cont pe baza de email

**Descriere:** Utilizatorul trebuie sa își poată crea un cont cu ajutorul unei adrese de email valide și a unei parole de cel puțin șase caractere.

#### Scenariu 1:

Pasul 1: Oprește conexiunea la Internet

Pasul 2: Intră în aplicație

Pasul 3: Da tap pe "Nu ai cont? Creează cont gratuit!"

Pasul 4: Apasă "Creează cont"

**Rezultat așteptat:** Un mesaj de eroare care să sugereze că nu este conexiune la Internet.

#### Scenariu 2:

Pasul 1: Pornește conexiune la Internet

Pasul 2: Intră în aplicație

Pasul 3: Da tap pe "Nu ai cont? Creează cont gratuit!"

Pasul 4: Apasă "Creează cont"

**Rezultat așteptat:** Un mesaj de eroare care să sugereze completarea câmpurilor goale.

#### Scenariu 3:

Pasul 1: Intră în aplicație

Pasul 2: Fă swipe dreapta pentru a ajunge la creează cont

Pasul 3: Introdu o adresă de email invalidă și lasă restul câmpurilor goale

Pasul 4: Apasă "Creează cont"

**Rezultat așteptat:** Un mesaj de eroare care să indice completarea unei adrese de email valide.

#### Scenariu 4:

Pasul 1: Intră în aplicație
Pasul 2: Da tap pe "*Nu ai cont? Creează cont gratuit!*"
Pasul 3: Introdu o adresă de email validă și restul câmpurilor goale
Pasul 4: Apasă "*Creează cont*"

**Rezultat așteptat:** Un mesaj de eroare care să sugereze completarea câmpurilor goale.

#### Scenariu 5:

Pasul 1: Intră în aplicație

Pasul 2: Da tap pe "Nu ai cont? Creează cont gratuit!"

**Pasul 3:** Introdu o adresă de email validă și o parolă cu mai puțin de 6 caractere iar restul câmpurilor sunt lăsate goale

Pasul 4: Apasă "Creează cont"

**Rezultat așteptat:** Un mesaj de eroare care să sugereze introducerea unei parole de cel puțin 6 caractere.

#### Scenariu 6:

Pasul 1: Intră în aplicație

Pasul 2: Da tap pe "Nu ai cont? Creează cont gratuit!"

**Pasul 3:** Introdu o adresă de email validă, o parola cu cel puțin 6 caractere și introdu în al doilea câmp de parolă o parolă care să nu fie egală cu prima parolă.

Pasul 4: Apasă "Creeaza cont"

**Rezultat așteptat:** Un mesaj de eroare care să sugereze confirmarea parolei (reintroducerea primei parole).

#### Scenariu 7:

Pasul 1: Intră în aplicație

Pasul 2: Da tap pe "Nu ai cont? Creează cont gratuit!"

**Pasul 3:** Introdu o adresă de email validă, o parolă cu cel puțin 6 caractere și în al doilea câmp de parolă reintrodu prima parola.

Pasul 4: Apasă "Creează cont"

**Rezultat așteptat:** Un mesaj care să sugereze crearea cu succes a contului și după doua secunde să apară pagina principala a aplicației.

TC2. Titlu: Descărcarea unui playlist

**Descriere:** Utilizatorul trebuie sa își poată descărca și salva un playlist în suportul de stocare al telefonului. Utilizatorul poate să oprească descărcarea, să repornească una oprită sau sa șteargă un playlist descărcat.

#### Scenariu 1:

Pasul 1: Pornește conexiunea la Internet dacă nu este, altfel sari la pasul 2

Pasul 2: Intră în aplicație

Pasul 3: Autentifica-te dacă nu ești autentificat, altfel sari la pasul 4

Pasul 4: Din meniul de pe pagina principala navighează la pagina cu playlist-uri

Pasul 5: Creează un playlist nou

Pasul 6: Navighează la pagina playlist-ului nou creat dând tap pe el

Pasul 7: Dă tap pe butonul de descărcare din bara de meniu

**Rezultat așteptat:** Un mesaj de eroare care sa sugereze că un playlist gol nu poate fi descărcat.

Scenariu 2:

Pasul 1: Pornește conexiunea la Internet daca nu este, altfel sari la pasul 2

Pasul 2: Intră în aplicație

Pasul 3: Autentifica-te dacă nu ești autentificat, altfel sari la pasul 4

**Pasul 4:** Din meniul de pe pagina principală navighează la pagina cu playlist-uri **Pasul 5:** Navighează la pagina unui playlist care conține cel puțin o melodie

Pasul 6: Da tap pe butonul de descărcare din bara de meniu

**Rezultat așteptat:** 1. În timp ce se descarcă o melodie, lângă aceasta trebuie sa apară o bara de progres a descărcării ei.

2. După ce se începe descărcarea butonul de descărcare din bara de meniu trebuie să devina unul de oprire a descărcării

2. Melodiile se descarcă secvențial (una după alta)

3. După ce s-a descărcat prima melodie, în colțul din stânga sus al imaginii albumului trebuie sa apară un triunghi portocaliu și să rămână așa până se descarcă tot playlistul. Când s-a descărcat tot playlist-ul triunghiul portocaliu trebuie să devină albastru.

4. După descărcarea unei melodii, pe imaginea aferenta acesteia trebuie să apară un triunghi albastru în colțul din stânga sus

#### Scenariu 3:

Pasul 1: Pornește conexiunea la Internet daca nu este, altfel sari la pasul 2

Pasul 2: Intră în aplicație

Pasul 3: Autentifica-te dacă nu ești autentificat, altfel sari la pasul 4

Pasul 4: Din meniul de pe pagina principală navighează la pagina cu playlist-uri

Pasul 5: Navighează la pagina unui playlist care conține cel puțin două melodii

Pasul 6: Dă tap pe butonul de descărcare din bara de meniu

**Pasul 7:** După ce se descarcă cel puțin o piesa, oprește descărcarea de la butonul de oprire din bara de meniu

## **Rezultat așteptat:** 1. Să se oprească descărcarea și piesa curentă să nu se salveze pe suportul de stocare

2. Triunghiul portocaliu al albumului sa fie vizibil

3. În partea de sus unde apar detaliile playlist-ului trebuie să apară cate melodii din total s-au descărcat

4. Triunghiul albastru trebuie să apară la fiecare melodie descărcata pana in momentul terminării descărcării

#### Scenariu 4:

Pasul 1: Pornește conexiunea la Internet daca nu este, altfel sari la pasul 2

Pasul 2: Intră în aplicație

Pasul 3: Autentifica-te dacă nu ești autentificat, altfel sari la pasul 4

Pasul 4: Din meniul de pe pagina principala navighează la pagina cu playlist-uri

Pasul 5: Navighează la pagina unui playlist care a fost descărcat parțial

Pasul 6: Dă tap pe butonul de continuare a descărcării din bara de meniu

Rezultat așteptat: 1. Descărcarea pieselor să continue de la ultima piesa descărcata

2. Triunghiul portocaliu să fie vizibil până la terminarea descărcării când îsi va schimba culoarea în albastru

3. La finalul descărcării butonul de oprire a descărcării din bara de meniu trebuie să devină buton de ștergere.

## Scenariu 5:

Pasul 1: Pornește conexiunea la Internet dacă nu este, altfel sari la pasul 2

Pasul 2: Intră în aplicație

Pasul 3: Autentifica-te dacă nu ești autentificat, altfel sari la pasul 4

Pasul 4: Din meniul de pe pagina principala navighează la pagina cu playlist-uri

Pasul 5: Navighează la pagina unui playlist care a fost descărcat

Pasul 6: Dă tap pe butonul de ștergere din bara de meniu

**Rezultat așteptat:** 1. Melodiile playlist-ului să fie șterse de pe suportul de stocare 2. Imaginea albumului și imaginea melodiilor nu trebuie să mai aibă nici un triunghi afișat.

#### Scenariu 6:

Pasul 1: Pornește conexiunea la Internet dacă nu este, altfel sari la pasul 2

Pasul 2: Intră în aplicație

Pasul 3: Autentifica-te dacă nu ești autentificat, altfel sari la pasul 4

**Pasul 4:** Din meniul de pe pagina principală navighează la pagina cu playlist-uri **Rezultat așteptat:** 1. Toate playlist-urile descărcate trebuie să aibă un triunghi

albastru în colțul stânga sus al imaginii playlist-ului

2. Toate playlist-urile descărcate parțial trebuie să aibă un triunghi portocaliu în colțul stânga sus al imaginii playlist-ului

## Scenariu 7:

Pasul 1: Pornește conexiunea la Internet dacă nu este, altfel sari la pasul 2

Pasul 2: Intră în aplicație

Pasul 3: Autentifica-te dacă nu ești autentificat, altfel sari la pasul 4

Pasul 4: Din meniul de pe pagina principală navighează la pagina cu playlist-uri

Pasul 5: Navighează la pagina unui playlist nedescărcat

Pasul 6: Dă tap pe butonul de descărcare

Pasul 7: Navighează înapoi la pagina cu playlist-uri

**Rezultat așteptat:** 1. Playlist-ul curent în momentul descărcării trebuie să aibă un triunghi portocaliu în coltul stânga sus al imaginii playlist-ului, iar în momentul terminării descărcării, triunghiul trebuie să își schimbe culoarea in albastru.

2. În timp ce playlist-ul se descarcă, în dreptul acestuia trebuie să apară o bară de progres care să indice progresul descărcării

## Capitolul 7. Manual de Instalare și Utilizare

Acest capitol reprezintă o descriere detaliată a modului de instalarea a aplicației pe telefonul mobil și definește pașii pe care trebuie să îi parcurgă utilizatorul pentru a putea înțelege și folosi aplicația cu succes.

## 7.1. Instalarea aplicației

Instalarea aplicației pe dispozitivul mobil se poate face în două moduri. În continuare voi descrie cele două moduri de instalare a aplicației.

## 7.1.1. Instalarea din Windows Store

Pentru instalarea aplicației din Windows Store pe dispozitivul mobil este necesar ca dispozitivul mobil să aibă conexiune la Internet și regiunea telefonului să fie setată pe România. Dacă condițiile enumerate mai sus sunt satisfăcute trebuie executați următorii pași:

- 1. Deschideți Windows Store de pe telefon
- 2. Căutați "Zonga"
- 3. Instalați aplicația apăsând butonul install

## 7.1.2. Instalarea aplicație folosind unealta Application Deployment

Pentru instalarea aplicației având fișierul cu extensia XAP descărcat, trebuie urmați următorii pași:

- 1. Instalați Windows Phone 8 SDK
- 2. Conectați telefonul mobil la calculator prin intermediul unul cablu USB
- 3. Porniți aplicația Application Deployment instalată odată cu Windows Phone 8 SDK. Ecranul principal al aplicație arată ca în figura 7.1
- 4. Din drop-down-ul Target selectați meniul Device
- 5. Dați click pe Browse și selectați fișierul cu extensia XAP
- 6. Apăsați butonul Deploy pentru instalarea aplicației pe dispozitivul mobil

₽	Application Deployment	×		
Application Deployment		dows Phone		
This tool allows you to install a prepackaged XAP on a registered Windows Phone. Please select the device target for installation and the XAP to be installed, and click "Deploy"				
Target:	Device v			
XAP: Status:	Device Emulator WVGA 512MB Emulator WVGA Emulator VXGA Emulator 720P Emulator 7.1 26MB Emulator 7.1 Emulator 7.8 Emulator 7.8	Browse		
		Deploy		

Figura 7.1 Ecranul principal al aplicației Application Deployment

## 7.2. Manual de utilizare

## 7.2.1. Pagina de autentificare și creare cont

La deschiderea aplicației în cazul în care un utilizator nu a mai fost niciodată autentificat va apărea pagina din figura 7.2. În cazul în care un utilizator nu are un cont în cadrul sistemului acesta va putea să își creeze unul nou dând swipe stânga sau dreapta pe pagină de autentificare și îi va părea secțiunea din figura 7.3 Dacă un utilizator a mai fost autentificat în cadrul aplicației dar nu este autentificat acesta va avea câmpurile în care trebuie să introducă numele utilizatorului și parola pre completate.



Figura 7.2 Pagina principală



Figura 7.3 Creează Cont



Figura 7.4 Credențiale pre completate

## 7.2.2. Pagina principală

La deschiderea aplicației dacă un utilizator este autentificat atunci acesta va vedea că prima pagină, pagina principală a aplicației. Meniul principal al aplicației este cel ilustrat în figura 7.5 cu ajutorul căruia se poate ajunge la paginile care îl interesează pe utilizator din cadrul aplicației. Făcând swipe stânga pe pagină principala se va ajunge fie la secțiunea "Now playing", fie la secțiunea "Ce e nou". Secțiunea "Now playling" va apărea doar dacă utilizatorul a ascultat cel puțin o melodie înainte de a părăsi aplicația sau dacă ascultă o melodie în momentul de fată. Dacă ce-a de-a două secțiune este "Now playing" atunci ce-a de-a treia este "Ce e nou".

În secțiunea "Now playing" la revenirea în aplicație va apărea ultima melodie pe care utilizatorul a ascultat-o înainte să închidă aplicația. Așa cum este ilustrat în figura 7.6 în această secțiune utilizatorul poate să vadă imaginea, numele și artistul melodiei. Pe lângă acestea utilizatorul are la dispoziție câteva controale cu ajutorul cărora poate să treacă la următoarea melodie sau la melodia anterioară, poate să pornească redarea unei melodii sau să pună pauză. Tot în această secțiune se poate vedea și progresul redării unei melodii.

În cadrul secțiunii "Ce e nou" se pot vizualiza cele mai importante noutăți referitoare la artiști, albume sau melodii din cadrul platformei așa cum este ilustrat în figura 7.7.



Figura 7.5 Meniul de pe pagina principală



Figura 7.6 Melodia care a fost ascultată ultima dată



Figura 7.7 Noutățile din cadrul platformei Zonga

## 7.2.3. Pagina cu recomandări

În cadrul acestei pagini se pot vizualiza melodiile, albumele, artiști și playlist-urile recomandate pentru fiecare utilizator pe baza istoricului său în cadrul sistemului Zonga. În cadrul acestor secțiuni va apărea o inimioară roșie dacă piesă, albumul sau artistul au fost categorisite ca fiind printre favoritele utilizatorului. Dacă se tine apăsat pe un element din aceste categorii, va apărea un meniu cu ajutorul căruia se poate adaugă un element în categoria favorite, se poate naviga la pagina artistului, se poate naviga la pagina albumului, se poate salva un playlist ca fiind al utilizatorului s.a.m.d. În cadrul fiecărei secțiuni se



Figura 7.8 Secțiunea melodii din recomandări



Figura 7.9 Secțiunea albume din recomandări



încarcă doar atâtea elemente de câte are nevoie utilizatorul și niciodată nu se vor încarca prea multe pentru a face un trafic de date prea mare.

Dacă se da tap pe o melodie, toate melodiile încărcate până acum vor fi adăugate în quicklist și va începe să fie redată melodia pe care s-a dat tap. În momentul în care se radă o melodie, în colțul dreapta sus al paginii va apărea un buton cu ajutorul căruia se poate ajunge la o pagină unde se gestionează melodiile care urmează a fi redate. Pagina pentru gestionarea melodiilor va fi descrisă mai jos.

Dacă se da tap pe un album, artist sau playlist se va ajunge la pagina albumului, artistului sau playlist-ului. Toate paginile menționate anterior vor fi descrise în cadrul altei secțiuni a acestui capitol.

## 7.2.4. Pagina cu topuri

În cadrul acestei pagini se pot vizualiza melodiile, albumele și artiști care sunt cei mai ascultați de către toți utilizatorii în cadrul sistemului Zonga. În cadrul acestor secțiuni va apărea o inimioară roșie dacă melodiile, albumul sau artistul au fost categorisite de către utilizator că făcând parte printre favoritele lui. Dacă se tine apăsat pe un element din aceste categorii va fi afișat un meniu prin intermediul căruia se poate adăuga un element în categoria favorite, se poate naviga la pagina artistului, se poate naviga la pagina albumui, se poate adăuga o piesă la un playlist al utilizatorului s.a.m.d. În cadrul fiecărei secțiuni se încarcă doar atâtea elemente câte are nevoie utilizatorul pentru a nu face un trafic prea mare de date.

Dacă se da tap pe o melodie, toate melodiile încărcate până acum vor fi adăugate în quicklist și va începe să fie redată melodia pe care s-a dat tap. În momentul în se redă o melodie, în colțul dreapta sus al paginii va apărea un buton cu ajutorul căruia se poate ajunge la pagina unde se gestionează melodiile care urmează să fie redate. Dacă se da tap pe un album sau artist se va ajunge la pagina albumului sau artistului.

Secțiunile paginii pot fi observate în cadrul figurilor 7.11, 7.12 și 7.13.





Figura 7.11 Secțiunea melodii din topuri



Figura 7.12 Secțiunea albume din topuri



Figura 7.13 Secțiunea artiști din topuri

## 7.2.5. Pagina cu elementele favorite

În cadrul aceste pagini se pot vizualiza melodiile, albumele și artiștii favoriți al utilizatorului autentificat. Dacă se tine apăsat pe un element din aceste categorii va apărea un meniu prin intermediul căruia se poate naviga la pagina albumului, se poate naviga la pagina artistului, se poate șterge un element din categoria favorite s.a.m.d.

În cadrul fiecărei secțiuni se încarcă doar atâtea elemente de câte are nevoie utilizatorul pentru a nu face un trafic prea mare de date.

Dacă se da tap pe o melodie, toate melodiile încărcate până acum vor fi adăugate în quicklist și va începe să fie redată melodia pe care s-a dat tap. În momentul în care este redată o melodie, în colțul dreapta sus al paginii va apărea un buton cu ajutorul căruia se poate ajunge la pagina unde se gestionează melodiile ce urmează să fie redate. Dacă se da tap pe un album sau un artist se va ajunge la pagină albumului sau artistului.

Secțiunile paginii favorite sunt ilustrare în figurile 7.14, 7.15 și 7.16.



## 7.2.6. Pagina playlist-urilor

În cadrul acestei pagini se pot vizualiza și gestiona toate playlist-urile utilizatorului autentificat. Dacă se da tap pe un playlist se va naviga la pagina unui playlist unde se pot vizualiza toate melodiile pe care le conține și câteva informații mai detaliate precum câte piese conține, ce durata are playlist-ul etc.

În cadrul paginii uni playlist se poate gestiona stocarea sa în telefonul mobil. În funcție de starea playlist-ului curent vor apărea diferite butoane pentru a descarcă, a rezuma sau a șterge playlist-ul din suportul de stocare local.

În timpul descărcării unui playlist se poate observa progresul descărcării fiecare melodii pe pagina unui playlist sau progresul total al descărcării în pagina cu toate playlisturile. În colțul stânga sus al unui playlist apare un triunghi portocaliu dacă playlist-ul este descărcat parțial sau este în curs de descărcare, sau un triunghi albastru dacă playlist-ul a fost descărcat complet pe dispozitiv. Pe lângă aceste indicatoare pe care le conține un

playlist, melodiile din cadrul unui playlist-ului vor conține aceleași tipuri de indicatoare: triunghi albastru dacă melodia este descărcată complet sau triunghi portocaliu dacă melodia este în curs de descărcare.

Toate funcționalitățile amintite mai sus sunt ilustrate în figurile 7.17, 7.18 și 7.19.



Figura 7.20 Pagina artistului, secțiunea melodii

Figura 7.21 Pagina artistului, sectiunea albume

Figura 7.22 Pagina artistului, sectiunea artisti similari

0

În cadrul acestei pagini se pot vizualiza cele mai importante melodii ale unui artist, albumele sale dar si un număr de artisti similari cu artistul curent. În figura 7.20 este ilustrată sectiunea cu melodiile unui artist. În cadrul acestei sectiuni este un buton care adaugă sau șterge artistul din categoria favorite. Dacă să da tap pe o melodie, toate melodiile afișate vor fi adăugate în quicklist și va începe redarea melodiei pe care s-a dat tap.

Secțiunea "albume" a paginii conține toate albumele artistului curent. Dacă se dă tap pe un album se navighează la pagina albumului.

Secțiunea "artiști similari" ilustrată în figura 7.22 conține un set de artiști similari cu artistul curent care s-ar putea să îl intereseze pe utilizator.

Pagina artistului are și ea butonul din dreapta sus care duce utilizatorul la pagina care gestionează melodiile ce urmează să fie ascultate.

## 7.2.8. Pagina unui album

Pe această pagină sunt afișate informații mai detaliate cu privire la un album. Pagina conține două secțiuni: secțiunea "melodii" și secțiunea "similare".

În secțiunea "melodii" ilustrată în figura 7.23 sunt afișate toate melodiile unui album. La acțiunea de "tap and hold" va apărea un meniu cu diferite opțiuni precum adăugarea la favorite, navigarea la pagina artistului etc. Dând tap pe o melodie această va fi adăugată în quicklist și va începe redarea ei.

Secțiunea "similare" conține un set de albume similare cu albumul curent. Această secțiune este ilustrată în imaginea 7.24.

În partea de jos a paginii se poate observa un meniu cu ajutorul căruia se adaugă sau se șterge albumul la categoria favorite și un buton pentru navigare la pagina artistului.



Figura 7.23 Pagina albumului, secțiunea melodii



Figura 7.24 Pagina albumului, secțiunea albume similare

## 7.2.9. Căutare

Aplicația oferă utilizatorilor posibilitatea de a caută mai rapid ceea ce îi interesează pe aceștia. În momentul introducerii unui text în caseta de text de pe pagina principală se poate observa că aplicația va oferi sugestii pe baza textului introdus până în momentul respectiv așa cum se poate observa în figura 7.25.



Figura 7.25 Cautare rapida

În cazul în care utilizatorul nu este mulțumit de sugestii primite acesta poate opta pentru o căutare avansată dând tap pe butonul de căutare. După tap utilizatorului îi va fi afișată o pagină cu trei secțiuni: "melodii", "albume" și "artiști" așa cum se poate observa în figurile 7.26, 7.27 și 7.28. Această pagină este similară ca și funcționalități în ceea ce privește opțiunile pe care le are utilizatorul.



Figura 7.26 Pagina de căutare, secțiunea melodii

Figura 7.27 Pagina de căutare, secțiunea albume

Figura 7.28 Pagina de căutare, secțiunea artiști similari

## 7.2.10. Quicklist

Pagina quicklist-ului este pagina în care se gestionează melodiile care urmează să fie ascultate și modul în care acestea sunt ascultate. Se poate ajunge la această pagină dând tap pe imaginea secțiunii "Now playing" din cadrul paginii principale. Pagina este împărțită în două secțiuni: "now playing" și "quicklist".

În cadrul secțiunii "quicklist" se pot vizualiza toate melodiile dintre care urmează să fie ascultate. Secțiunea este ilustrată în figura 7.29.

Secțiunea "now playing" este asemănătoare cu secțiunea "now playing" de pe pagină principală doar că are mai multe opțiuni. De aici se poate selecta cum să fie redate melodiile: melodiile să fie redate aleatoriu, să se repete quicklist-ul sau să repete melodia curentă de un număr infinit de ori. Cu ajutorul barei de progress se poate ajunge la un anumit moment în cadrul melodiei dacă melodia permite acest lucru.

În partea dreapta a imaginii din cadrul secțiunii "now playing" așa cum este ilustrat în figura 2.30 se pot observa o serie de opțiuni precum navigarea la pagină artistului, navigare la pagină albumului, adăugarea melodiei într-un playlist sau adăugarea/ștergerea melodiei din cadrul categoriei favorite.

Trebuie specificat faptul că o melodie va fi redatată tot timpul de pe suportul de stocare al dispozitivului, dacă este descărcată, chiar dacă dispozitivul are acces la Internet pentru a nu face trafic utilizatorului.



Figura 7.29 Pagina quicklist, secțiunea quicklist



Figura 7.30 Pagina quicklist, secțiunea now playing

## Capitolul 8. Concluzii și dezvoltări ulterioare

În acest capitol se vor prezenta obiectivele atinse, modalitatea în care a fost promovată aplicația și câteva dezvoltări ulterioare

## 8.1. Objective realizate

Prin intermediul acestui proiect dezvoltatorul aplicației a dobândit cunoștințe noi și a pus în practică noțiunile învățate pe parcursul studiilor superioare.

Pentru dezvoltarea aplicației dezvoltatorul a parcurs următoarele etape:

- A studiat piața și a speculat oportunitățile care pot fi integrate în cadrul aplicației
- A studiat și a dobândit cunoștințe noi despre platforma Windows Phone 8, standarde pentru fișiere media și diverse tehnici pentru streaming de date
- A realizat analiza sistemului, a detectat tehnologiile care au fost utilizate pentru dezvoltarea sistemului și a creat un plan de proiect pentru a avea o viziune de ansamblu cât mai corectă și mai exactă asupra evoluției proiectului
- A ales să dezvolte produsul software într-o manieră iterativă bazată pe metodologia SCRUM și a împărțit dezvoltarea în iterații de cate două săptămâni
- În cadrul fiecărei iterații a realizat analiza cerințelor care urmau să fie implementate, le-a implementat, a creat scenarii de test și a validat pe baza scenariilor de rest create funcționalitatea implementată. Pentru fiecare cerință din etapa de analiză a propus un set de soluții și a ales-o pe cea mai optimă dintre ele
- A dezvoltat componente speciale pentru respectarea cerințelor nonfuncționale ale aplicației

Conform planului de proiect întocmit, s-au realizat toate funcționalitățile aplicației care au fost propuse în capitolul 2 și anume:

- Autentificarea utilizatorilor
- Creare de cont nou
- Vizualizarea melodiilor, artiștilor, albumelor și playlist-urilor recomandate
- Vizualizarea melodiilor, albumelor și artiștilor din cadrul secțiuni de topuri
- Vizualizarea și efectuarea de operații de adăugare și ștergere a elementelor favorite
- Gestionarea playlist-urilor utilizatorului
- Salvarea unui playlist în mod offline
- Căutarea anumitor elemente pe baza unor cuvinte cheie
- Vizualizarea noutăților
- Redarea melodiilor de pe Internet sau de pe suportul de stocare al dispozitivului utilizatorului

## 8.2. Promovarea aplicației

Aplicația a fost publicată pe Windows Phone Store de unde utilizatorii platformei o pot descărca gratuit și pot beneficia de funcționalitățile sistemului Zonga. O altă modalitate de promovare a aplicației a fost prin intermediul campaniilor create de către dezvoltatorii sistemului Zonga pe pagina de Facebook a sistemului, prin intermediul unor afișe, postere și alte materiale de marketing.

Datorită acestor modalități de promovare prin care s-a făcut cunoscută aplicația, aceasta este utilizată frecvent de aproximativ o mie de utilizatori și a fost descărcată de peste trei mii de ori. Numărul mediu de utilizatori ai aplicației reprezintă aproximativ 7% din numărul total de utilizatori ai sistemului Zonga.

## 8.3. Dezvoltări ulterioare

Tot timpul vor exista îmbunătățiri care se vor putea aduce unei aplicații, însă cele mai prioritare care se vor dezvolta în versiunea următoare a aplicației sunt cele propuse de către utilizatori.

Dintre funcționalitățile viitoare se pot enumera:

- Introducerea unui mecanism de cache-ing pentru reducerea traficului
- Introducerea unui radio care să genereze automat un playlist pe baza preferințelor utilizatorului și a genului de muzică pe care dorește să îl asculte
- Reducerea consumului de energie
- Crearea unui mecanism de control al traficului care să îi permită utilizatorului să specifice traficul maxim care poate fi făcut de aplicație întro săptămână
- Posibilitatea de ascultare a melodiilor la calitate mare când utilizatorul dispune de conexiune la Internet prin standardul 3G sau 4G
- Vizualizarea activităților prietenilor utilizatorului de pe rețelele de socializare direct în cadrul aplicație

#### Bibliografie

- [1] J. Rivera şi R. v. d. Meulen, "Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones...," Gartner, 13 Februarie 2014.
   [Interactiv]. Available: http://www.gartner.com/newsroom/id/2665715.
   [Accesat 8 Decembrie 2013].
- [2] M. Jo Foley, "Microsoft's Windows Phone 8 finally gets a 'real' Windows core," ZDNet, 20 Iunie 2012. [Interactiv]. Available: http://www.zdnet.com/blog/microsoft/microsofts-windows-phone-8-finallygets-a-real-windows-core/12975. [Accesat 8 Noiembrie 2013].
- [3] P. Thurrott, Windows Phone 7 Secrets, Wiley Publishing, Inc., 2011.
- [4] P. Thurrot, Windows Phone 8 Field Guide, 2014.
- [5] "Isolated Storage," Microsoft, [Interactiv]. Available: http://msdn.microsoft.com/en-us/library/3ak841sy(v=vs.110).aspx. [Accesat 9 Septembrie 2013].
- [6] P. Kshitij, "Performance comparisons LinQ to SQL,ADO,C#," 26 Mai 2008. [Interactiv]. Available: http://www.codeproject.com/Articles/26367/Performance-comparisons-LinQ-to-SQL-ADO-C. [Accesat 11 Decembrie 2013].
- [7] J. Alva, "Potential Performance Issues with Compiled LINQ Query Re-Compiles," Microsoft Developer Network, 8 Iunie 2010. [Interactiv]. Available: http://blogs.msdn.com/b/appfabriccat/archive/2010/08/06/potential-performance-issues-with-compiled-linq-query-re-compiles.aspx?. [Accesat 11 Octombrie 2013].
- [8] S. Klein, Professional LINQ, Wiley Publishing, Inc., 2008.
- [9] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach şi T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," [Interactiv]. Available: http://www.w3.org/Protocols/HTTP/1.1/rfc2616.pdf.
- [10] J. Zeldman şi E. Marcotte, Designing with Web Standards, 3 ed., New Riders, 2009.
- [11] "Web Services Architecture," 11 Februarie 2004. [Interactiv]. Available: http://www.w3.org/TR/ws-arch/wsa.pdf. [Accesat 11 Octombrie 2013].
- [12] R. T. Fielding şi R. N. Taylor, ACM Transactions on Internet Technology, vol. 2, New York: Association for Computing Machinery, 2002, pp. 115-150.
- [13] J. Grossman, "Introduction to Model/View/ViewModel pattern for building WPF apps," 8 Octombrie 2005. [Interactiv]. Available: http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx. [Accesat 5 Noiembrie 2013].
- [14] G. Block, "Patterns For Building Composite Applications With WPF," *MSDN Magazine*, Septembrie 2008.

## Glosar de termeni

- API Application Programming Interface
- CPU Central Processing Unit
- DAO Data Access Unit
- HTTP HyperText Transfer Protocol
- HTTPS HyperText Transfer Protocol Secure
- JSON JavaScript Object Notation
- REST Representational State Transfer
- SDK Software Development Kit
- SOAP Simple Object Access Protocol
- URL Uniform Resource Locator
- XML Extensible Markup Language

## Listă tabele și figuri

Figura 3.1 Exemplu de cerere și răspuns prin intermediul API-ului	5
Figura 3.2 Exemplu de răspuns cu eroare	5
Figura 4.1 Diagrama Use Case pentru CF 1	2
Figura 4.2 Diagrama Use Case pentru CF 2	3
Figura 4.3 Diagrama Use Case pentru CF 3	4
Figura 4.4 Diagrama Use Case pentru CF 4 16	6
Figura 4.5 Diagrama Use Case pentru CF 5 17	7
Figura 4.6 Ecranul de start pentru Windows Phone 7	0
Figura 4.7 Opțiuni de stocare a datelor cu Isolated Storage și clasele aferente 22	2
Figura 4.8 Exemplu de cerere și răspuns HTTP	5
Figura 4.9 Folosirea verbelor HTTP pentru accesarea resurselor pe server 26	6
Figura 4.10 XML vs JSON	6
Figura 4.11 Clasele MVVM si interactiunea dintre ele	7
Figura 4.12 Diagrama UML a pattern-ului Abstract Factory	8
Figura 5.1 Arhitectura Client-Server a sistemului Zonga	9
Figura 5.2 Arhitectura aplicație Zonga server	1
Figura 5.3 Arhitectura aplicație Zonga pentru Windows Phone 8	2
Figura 5.4 Componentele utilizate pentru redarea melodiilor salvate local 35	5
Figura 5.5 Schema logică simplificată pentru redarea melodiilor 30	6
Figura 5.6 Componentele utilizate pentru redarea melodiilor de pe Internet 38	8
Figura 5.7 Diagrama de secvență pentru descărcarea unui playlist	9
Figura 7.1 Ecranul principal al aplicației Application Deployment	4
Figura 7.2 Pagina principală	5
Figura 7.3 Creează Cont	5
Figura 7.4 Credențiale pre completate	5
Figura 7.5 Meniul de pe pagina principală 40	6
Figura 7.6 Melodia care a fost ascultată ultima dată 46	6
Figura 7.7 Noutățile din cadrul platformei Zonga 40	6
Figura 7.8 Secțiunea melodii din recomandări 40	6
Figura 7.9 Secțiunea albume din recomandări 40	6
Figura 7.10 Secțiunea playlist-uri din recomandări 40	6

Figura 7.11 Secțiunea melodii din topuri	. 47
Figura 7.12 Secțiunea albume din topuri	. 47
Figura 7.13 Secțiunea artiști din topuri	. 47
Figura 7.14 Secțiunea melodii din favorite	. 48
Figura 7.15 Secțiunea albume din favorite	. 48
Figura 7.16 Secțiunea artiști din favorite	. 48
Figura 7.17 Pagina playlist-urilor	. 49
Figura 7.18 Pagina unui playlist	. 49
Figura 7.19 Pagina unui playlist descărcat	. 49
Figura 7.20 Pagina artistului, secțiunea melodii	. 49
Figura 7.21 Pagina artistului, secțiunea albume	. 49
Figura 7.22 Pagina artistului, secțiunea artiști similari	. 49
Figura 7.23 Pagina albumului, secțiunea melodii	. 50
Figura 7.24 Pagina albumului, secțiunea albume similare	. 50
Figura 7.25 Cautare rapida	. 51
Figura 7.26 Pagina de căutare, secțiunea melodii	. 51
Figura 7.27 Pagina de căutare, secțiunea albume	. 51
Figura 7.28 Pagina de căutare, secțiunea artiști similari	. 51
Figura 7.29 Pagina quicklist, secțiunea quicklist	. 52
Figura 7.30 Pagina quicklist, secțiunea now playing	. 52

Tabel 3.1 Comparație între aplicațiile descrise și aplicația curentă	
Tabel 4.1 Cerințe funcționale	
Tabel 4.2 Cerințe Non-Funcționale	