US 20060195479A1

(19) **United States**

(12) **Patent Application Publication**

Spiegelman et al.

(10) **Pub. No.: US 2006/0195479 A1**

(43) **Pub. Date: Aug. 31, 2006**

(54) **METHOD FOR SHARING AND SEARCHING PLAYLISTS**

(76) Inventors: **Michael Spiegelman**, Los Angeles, CA (US); **David E. Brown**, Boston, MA (US); **Lori Ann Sutton**, La Jolla, CA (US); **Pierre Schiro**, Culver City, CA (US); **Christa Maria Stelzmuller**, Long Beach, CA (US); **Siddhartha S. Sahay**, Rolling Hills, CA (US); **Daniel Baird**, Carlsbad, CA (US); **Adam Korman**, Glendale, CA (US); **Alexander Stephen Toth III**, Venice, CA (US); **Poorna Chandra Gowda Bannikuppe Ramachandra**, Bangalore (IN); **Bipin Suresh**, Bangalore (IN)

Correspondence Address:
GREENBERG TRAURIG, LLP
MET LIFE BUILDING
200 PARK AVENUE
NEW YORK, NY 10166 (US)

(21) Appl. No.: **11/315,419**

(22) Filed: **Dec. 21, 2005**

Related U.S. Application Data

(60) Provisional application No. 60/657,222, filed on Feb. 28, 2005. Provisional application No. 60/678,718, filed on May 5, 2005.

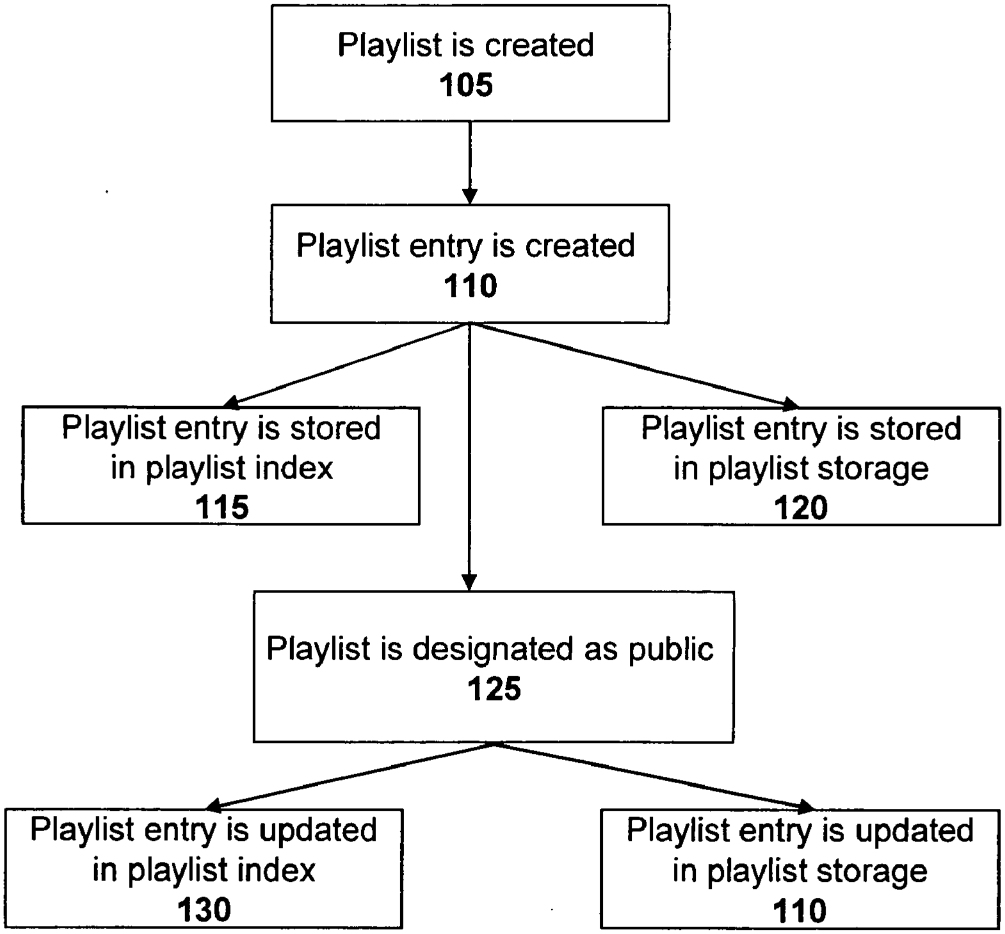
Publication Classification

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **707/104.1**

(57) **ABSTRACT**

The present disclosure provides a method for making a playlist available to the public. In another embodiment, the present disclosure provides a method for creating a playlist comprising user-defined descriptor information. In another embodiment, the present disclosure provides a method for searching for such public playlists.


Playlist is created
105
Playlist entry is created
110
Playlist entry is stored in playlist index
115
Playlist entry is stored in playlist storage
120
Playlist is designated as public
125
Playlist entry is updated in playlist index
130
Playlist entry is updated in playlist storage
110

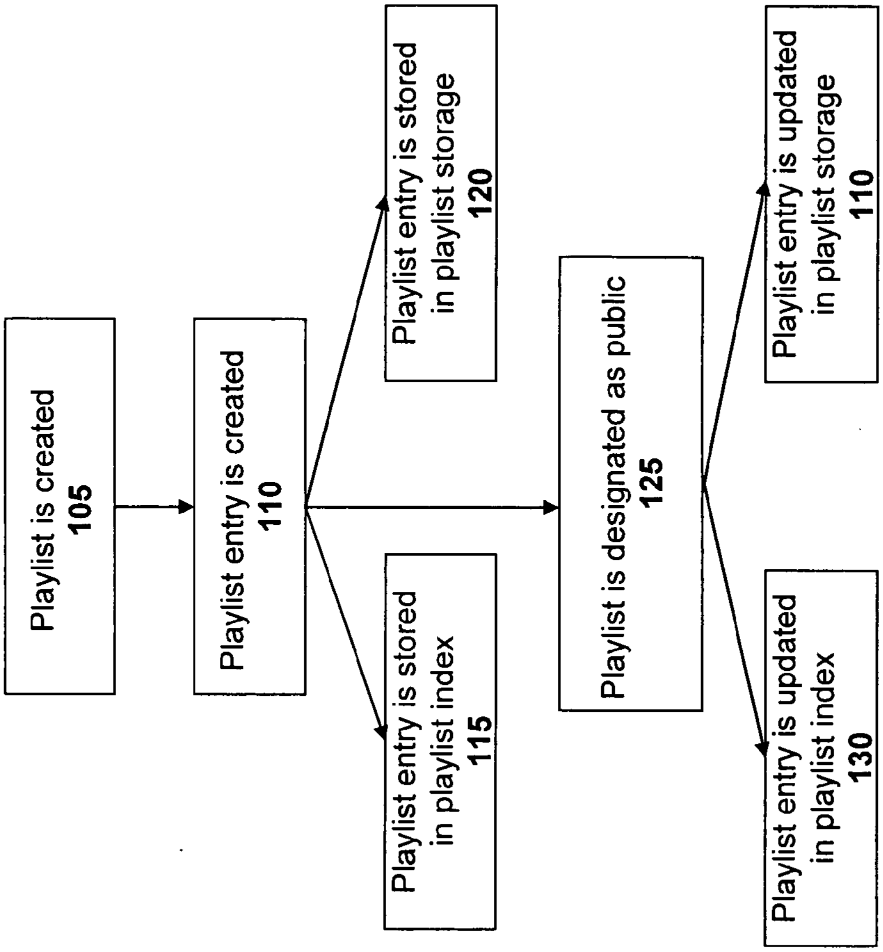
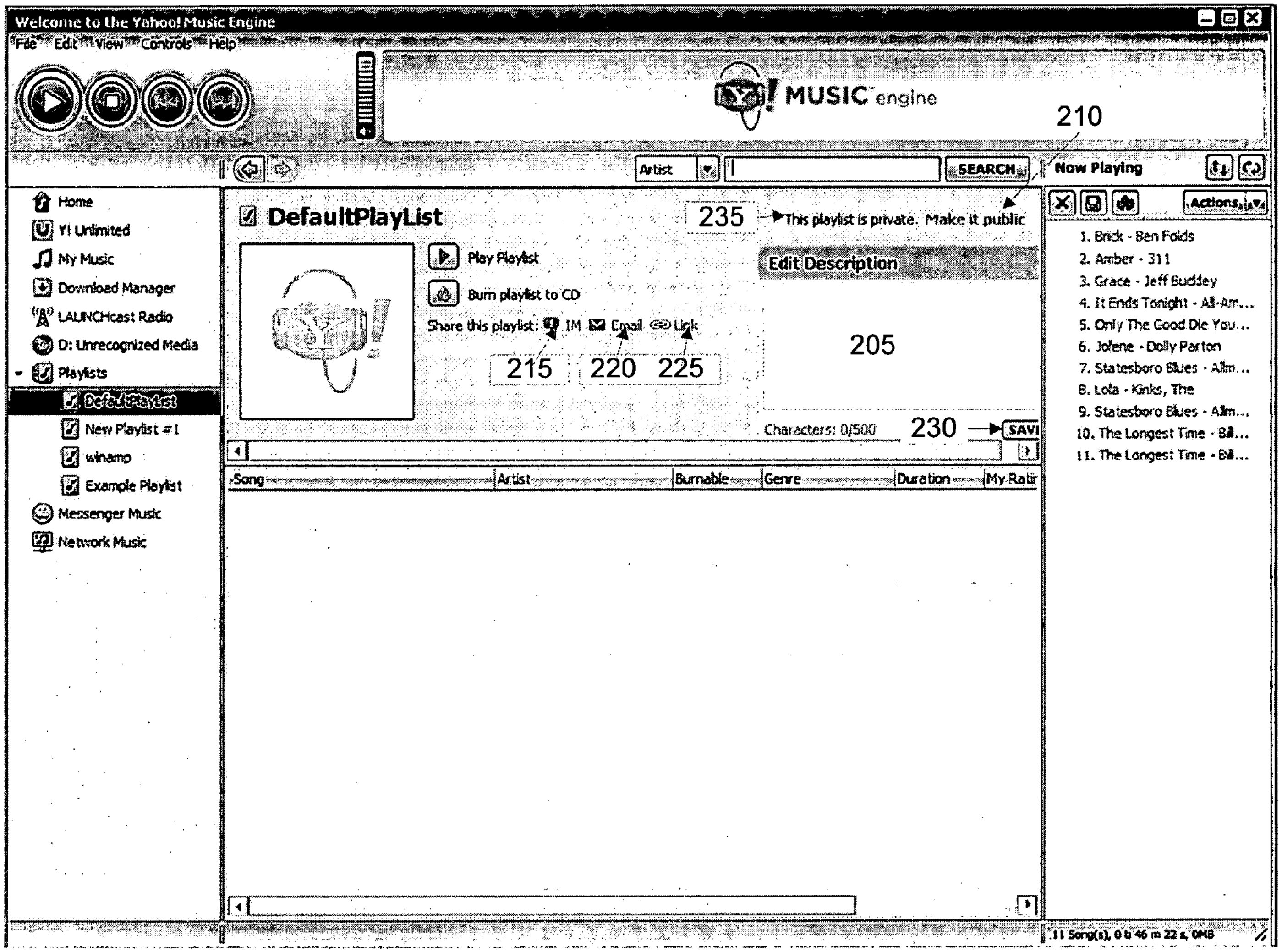

Playlist is created
105
Playlist entry is created
110
Playlist entry is stored in playlist index
115
Playlist entry is stored in playlist storage
120
Playlist is designated as public
125
Playlist entry is updated in playlist index
130
Playlist entry is updated in playlist storage
110

FIG. 1



This playlist is private. Make it public

Characters: 0/500

1. Brick - Ben Folds
2. Amber - 311
3. Grace - Jeff Buckley
4. It Ends Tonight - All-Am...
5. Only The Good Die You...
6. Jolene - Dolly Parton
7. Statesboro Blues - Allm...
8. Lola - Kinks, The
9. Statesboro Blues - Allm...
10. The Longest Time - Bil...
11. The Longest Time - Bil...

11 Song(s), 0 h 46 m 22 s, 0MB

FIG. 2

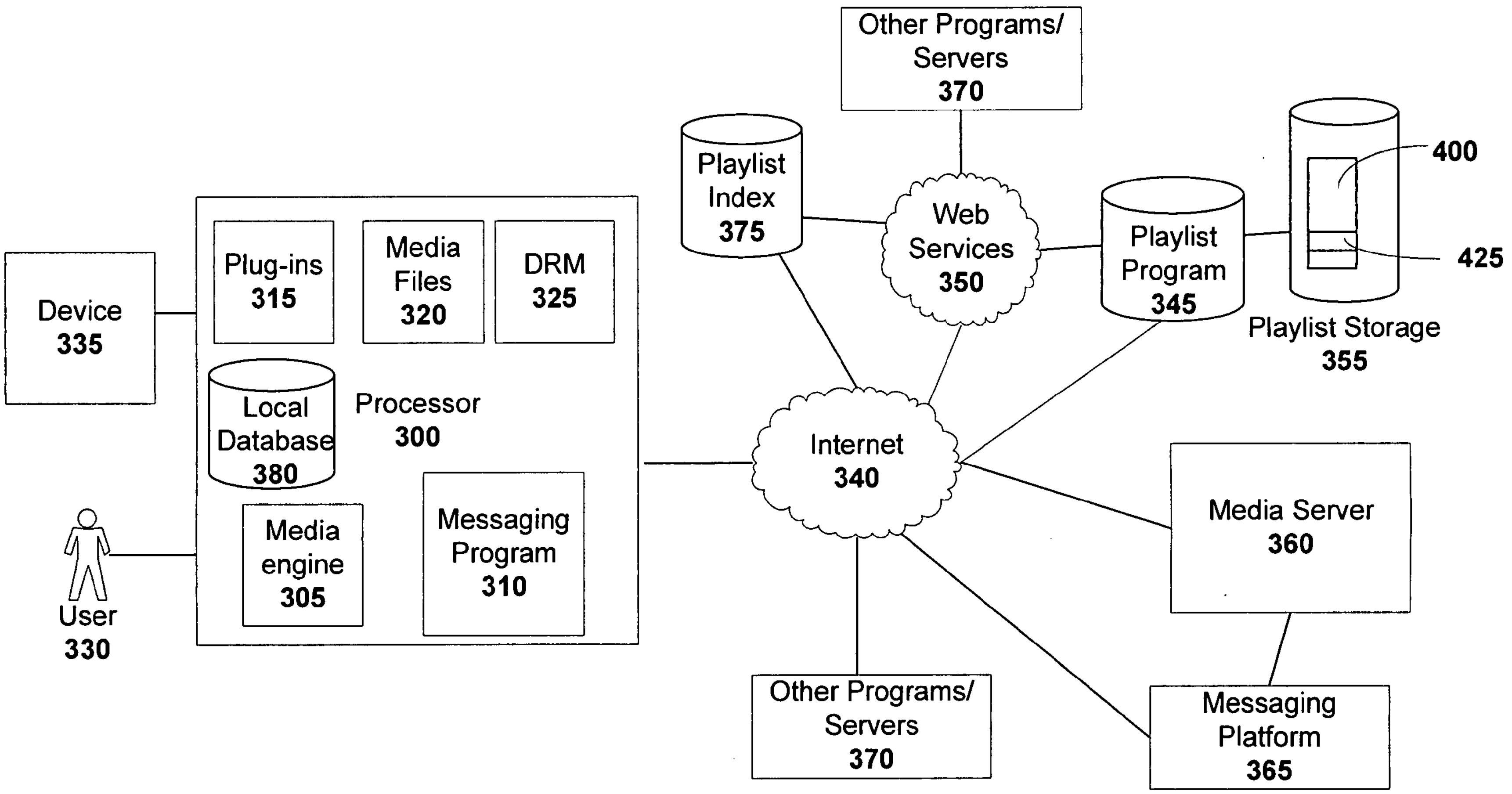

Other Programs/
Servers
370
Playlist
Index
375
Web
Services
350
Playlist
Program
345
400
425
Playlist Storage
355
Device
335
Plug-ins
315
Media
Files
320
DRM
325
Local
Database
380
Processor
300
Internet
340
Media Server
360
Media
engine
305
Messaging
Program
310
User
330
Other Programs/
Servers
370
Messaging
Platform
365

FIG. 3

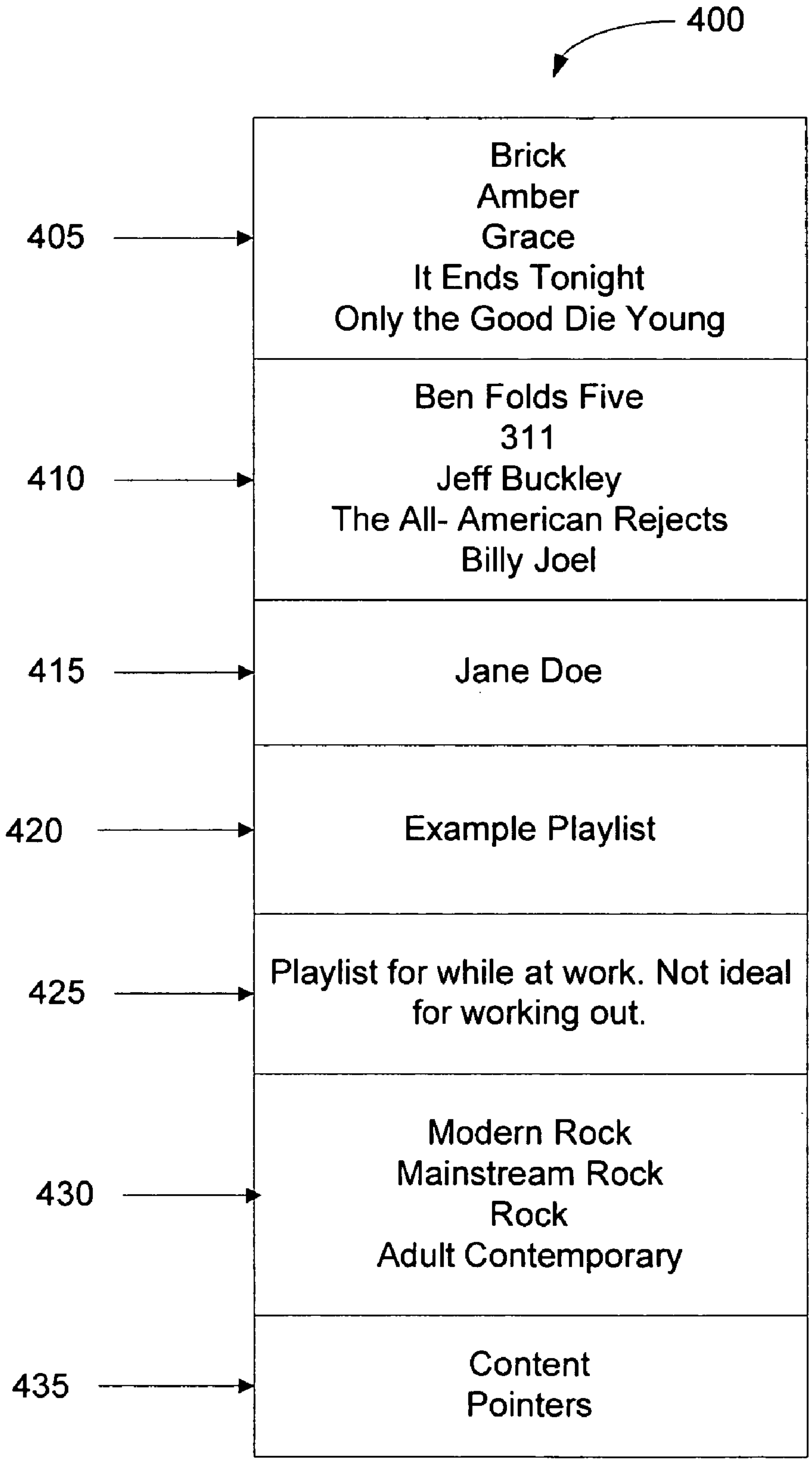
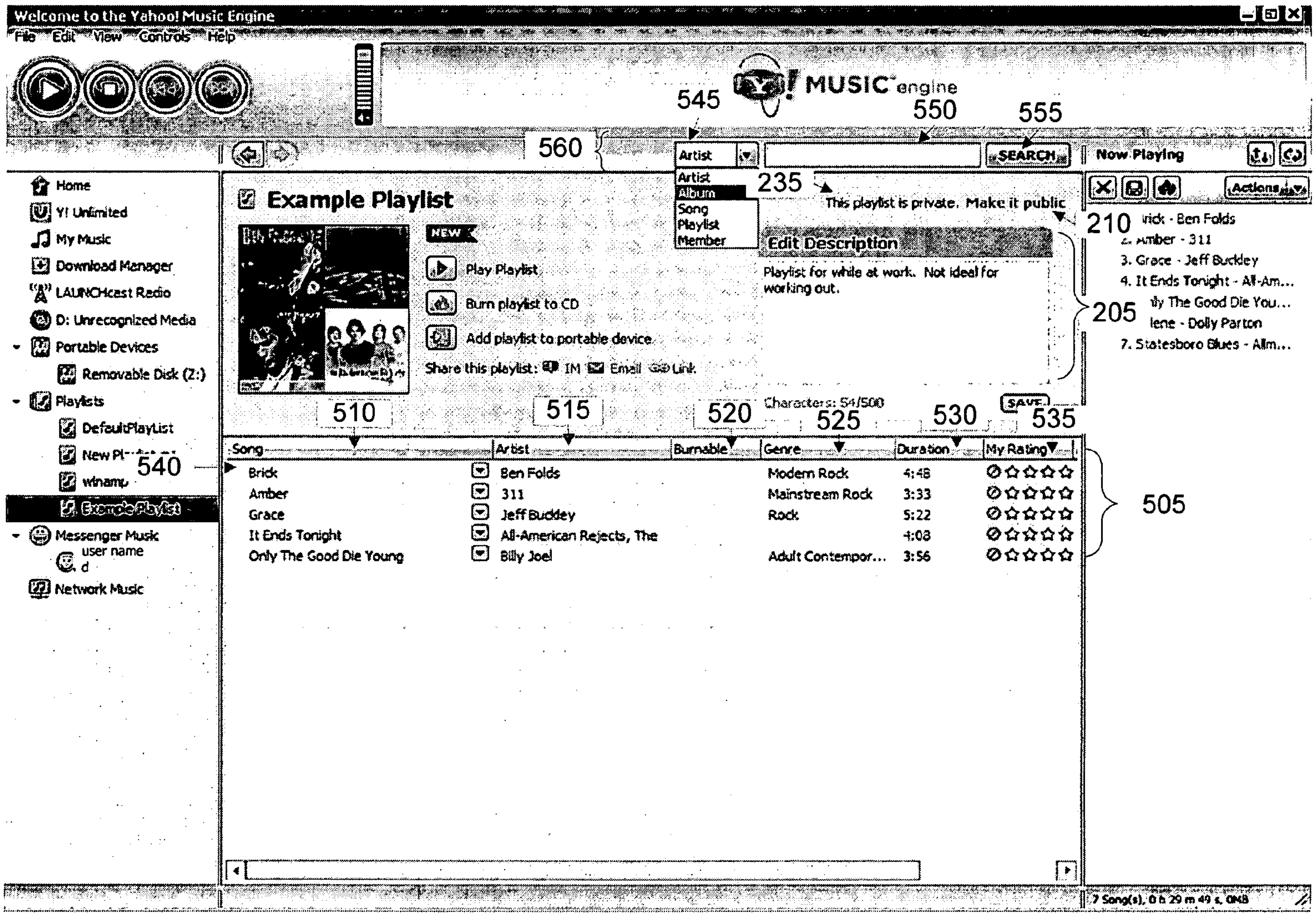

400
405
Brick
Amber
Grace
It Ends Tonight
Only the Good Die Young
410
Ben Folds Five
311
Jeff Buckley
The All- American Rejects
Billy Joel
415
Jane Doe
420
Example Playlist
425
Playlist for while at work. Not ideal for working out.
430
Modern Rock
Mainstream Rock
Rock
Adult Contemporary
435
Content
Pointers

FIG. 4



Example Playlist

This playlist is private. Make it public

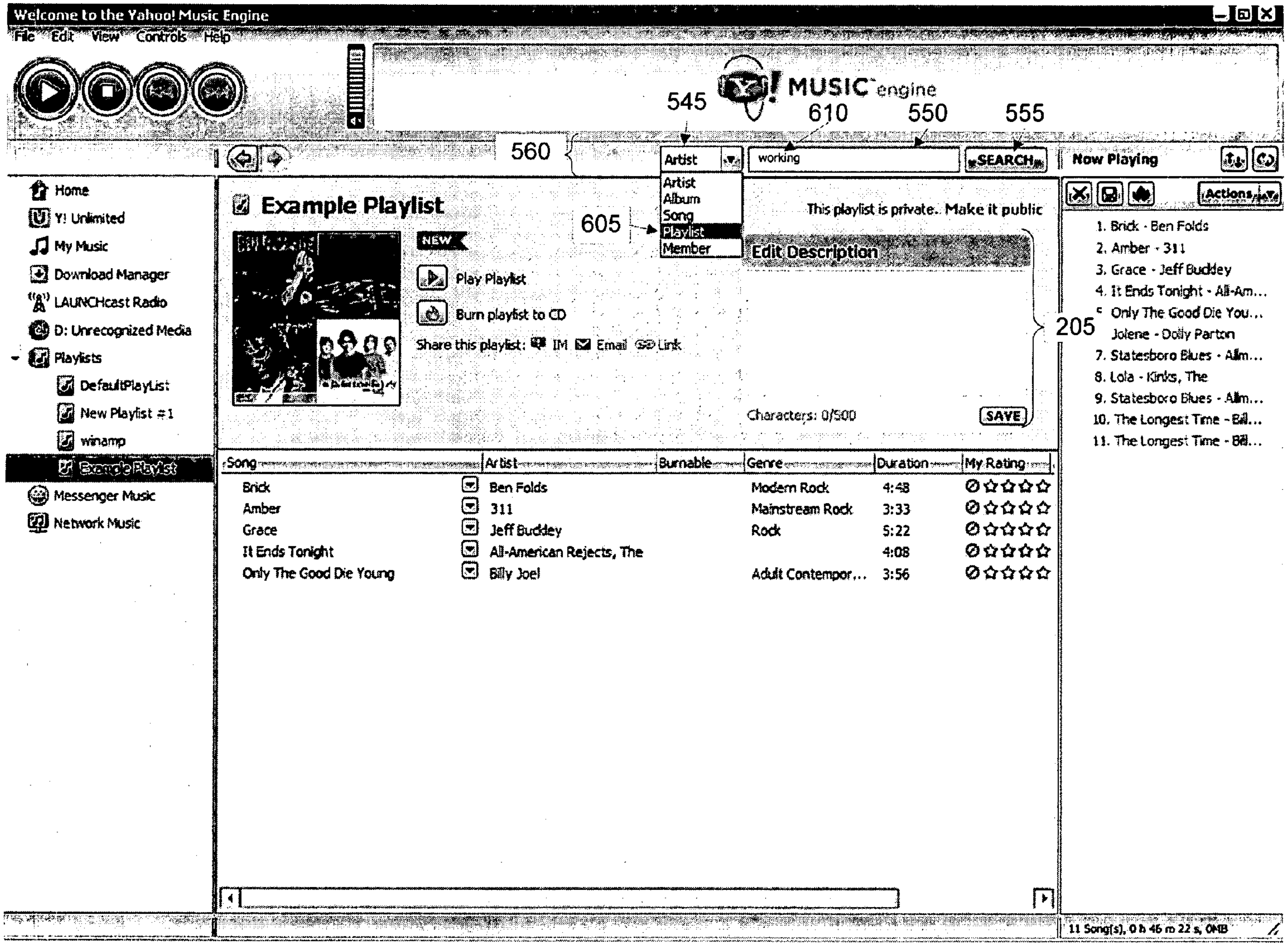
Playlist for while at work. Not ideal for working out.

Song	Artist	Burnable	Genre	Duration	My Rating
Brick	Ben Folds		Modern Rock	4:48	⊘☆☆☆☆☆
Amber	311		Mainstream Rock	3:33	⊘☆☆☆☆☆
Grace	Jeff Buckley		Rock	5:22	⊘☆☆☆☆☆
It Ends Tonight	All-American Rejects, The			4:08	⊘☆☆☆☆☆
Only The Good Die Young	Billy Joel		Adult Contempor...	3:56	⊘☆☆☆☆☆

rick - Ben Folds
. Amber - 311
3. Grace - Jeff Buckley
4. It Ends Tonight - All-Am...
ly The Good Die You...
lene - Dolly Parton
7. Statesboro Blues - Allm...

7 Song(s), 0 h 29 m 49 s, 0MB

FIG. 5



Example Playlist

This playlist is private.. Make it public

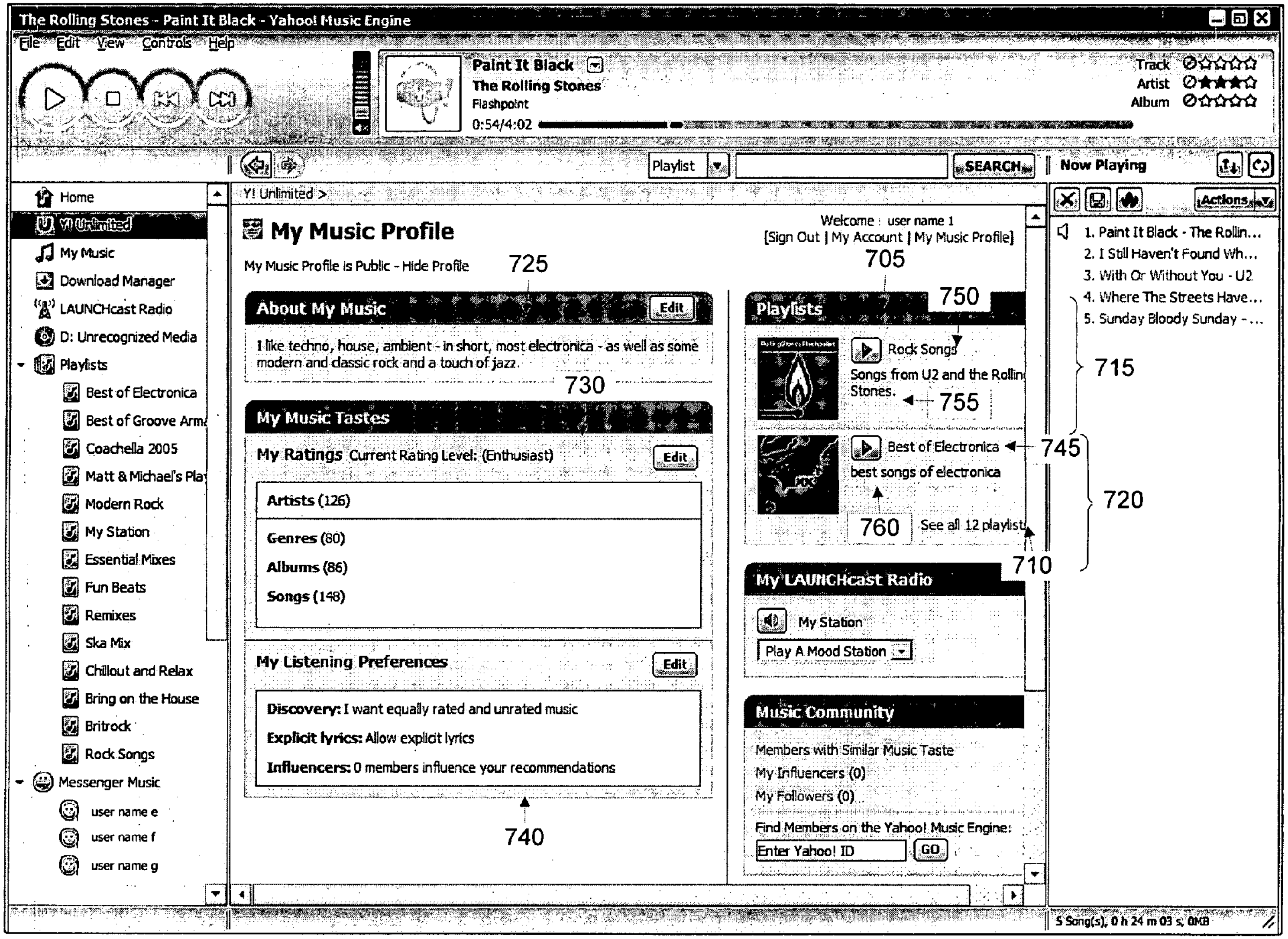
Characters: 0/500

Song	Artist	Burnable	Genre	Duration	My Rating
Brick	Ben Folds		Modern Rock	4:48	⊘☆☆☆☆☆
Amber	311		Mainstream Rock	3:33	⊘☆☆☆☆☆
Grace	Jeff Buckley		Rock	5:22	⊘☆☆☆☆☆
It Ends Tonight	All-American Rejects, The			4:08	⊘☆☆☆☆☆
Only The Good Die Young	Billy Joel		Adult Contempor...	3:56	⊘☆☆☆☆☆

1. Brick - Ben Folds
2. Amber - 311
3. Grace - Jeff Buckley
4. It Ends Tonight - All-Am...
5. Only The Good Die You...
6. Jolene - Dolly Parton
7. Statesboro Blues - Allm...
8. Lola - Kinks, The
9. Statesboro Blues - Allm...
10. The Longest Time - Bil...
11. The Longest Time - Bil...

11 Song(s), 0 h 46 m 22 s, 0MB

FIG. 6



The Rolling Stones - Paint It Black - Yahoo! Music Engine

File Edit View Controls Help

Paint It Black
The Rolling Stones
Flashpoint
0:54/4:02

Track
Artist
Album

Playlist SEARCH Now Playing

Home
Y! Unlimited
My Music
Download Manager
LAUNCHcast Radio
D: Unrecognized Media
Playlists
Best of Electronica
Best of Groove Arm
Coachella 2005
Matt & Michael's Pla
Modern Rock
My Station
Essential Mixes
Fun Beats
Remixes
Ska Mix
Chillout and Relax
Bring on the House
Britrock
Rock Songs
Messenger Music
user name e
user name f
user name g

Y! Unlimited >

My Music Profile

My Music Profile is Public - Hide Profile

Welcome : user name 1
[Sign Out | My Account | My Music Profile]

About My Music Edit

I like techno, house, ambient - in short, most electronica - as well as some modern and classic rock and a touch of jazz.

My Music Tastes

My Ratings Current Rating Level: (Enthusiast) Edit

Artists (126)

Genres (80)
Albums (86)
Songs (148)

My Listening Preferences Edit

Discovery: I want equally rated and unrated music
Explicit lyrics: Allow explicit lyrics
Influencers: 0 members influence your recommendations

Playlists

Rock Songs
Songs from U2 and the Rollin Stones.

Best of Electronica
best songs of electronica

See all 12 playlist

My LAUNCHcast Radio

My Station
Play A Mood Station

Music Community

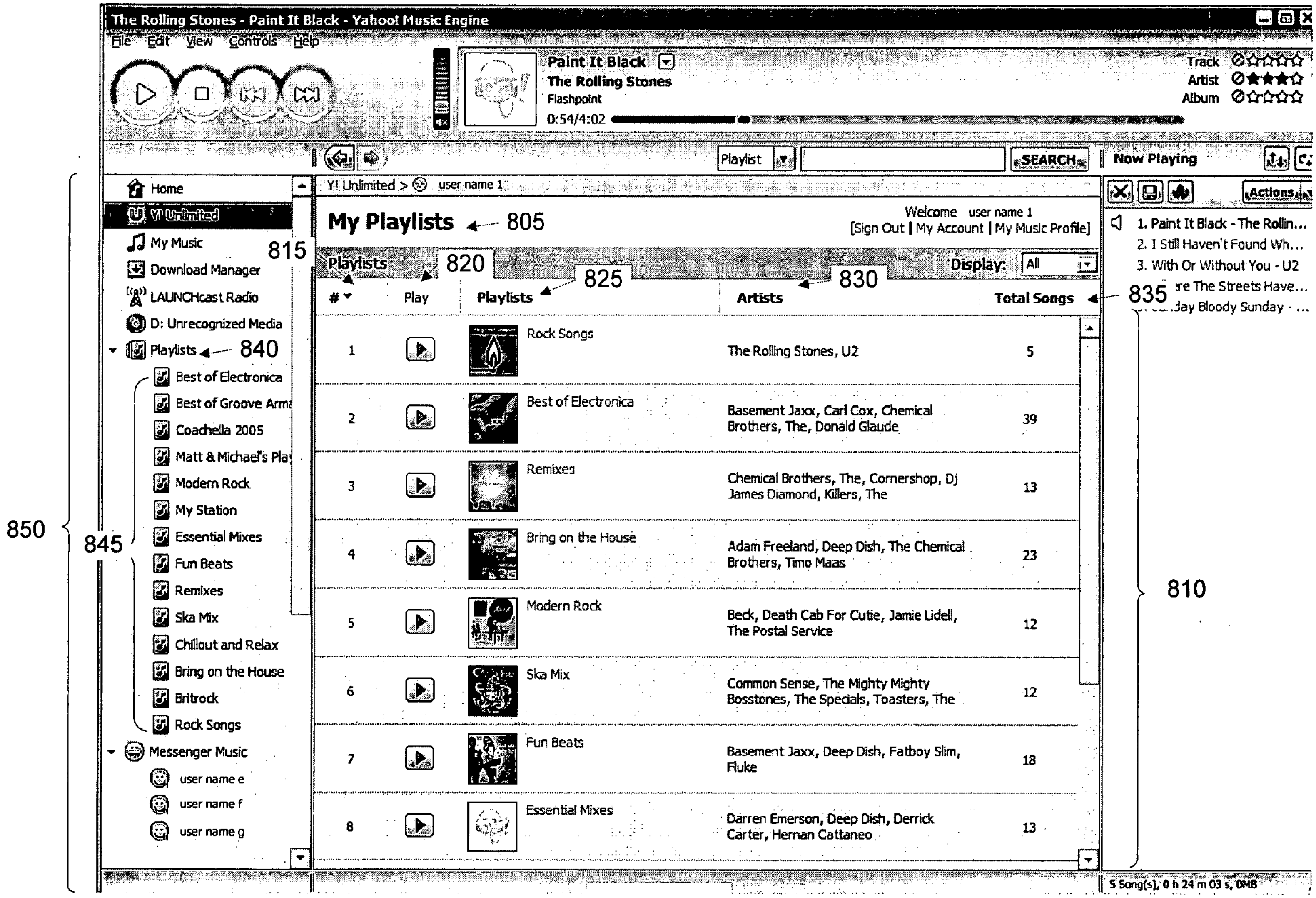
Members with Similar Music Taste
My Influencers (0)
My Followers (0)
Find Members on the Yahoo! Music Engine:
Enter Yahoo! ID GO

Actions

1. Paint It Black - The Rollin...
2. I Still Haven't Found Wh...
3. With Or Without You - U2
4. Where The Streets Have...
5. Sunday Bloody Sunday - ...

5 Song(s), 0 h 24 m 03 s, 0MB

FIG. 7



Paint It Black
The Rolling Stones
Flashpoint
0:54/4:02

Track
Artist
Album

Y! Unlimited > user name 1

My Playlists

Welcome user name 1
[Sign Out | My Account | My Music Profile]

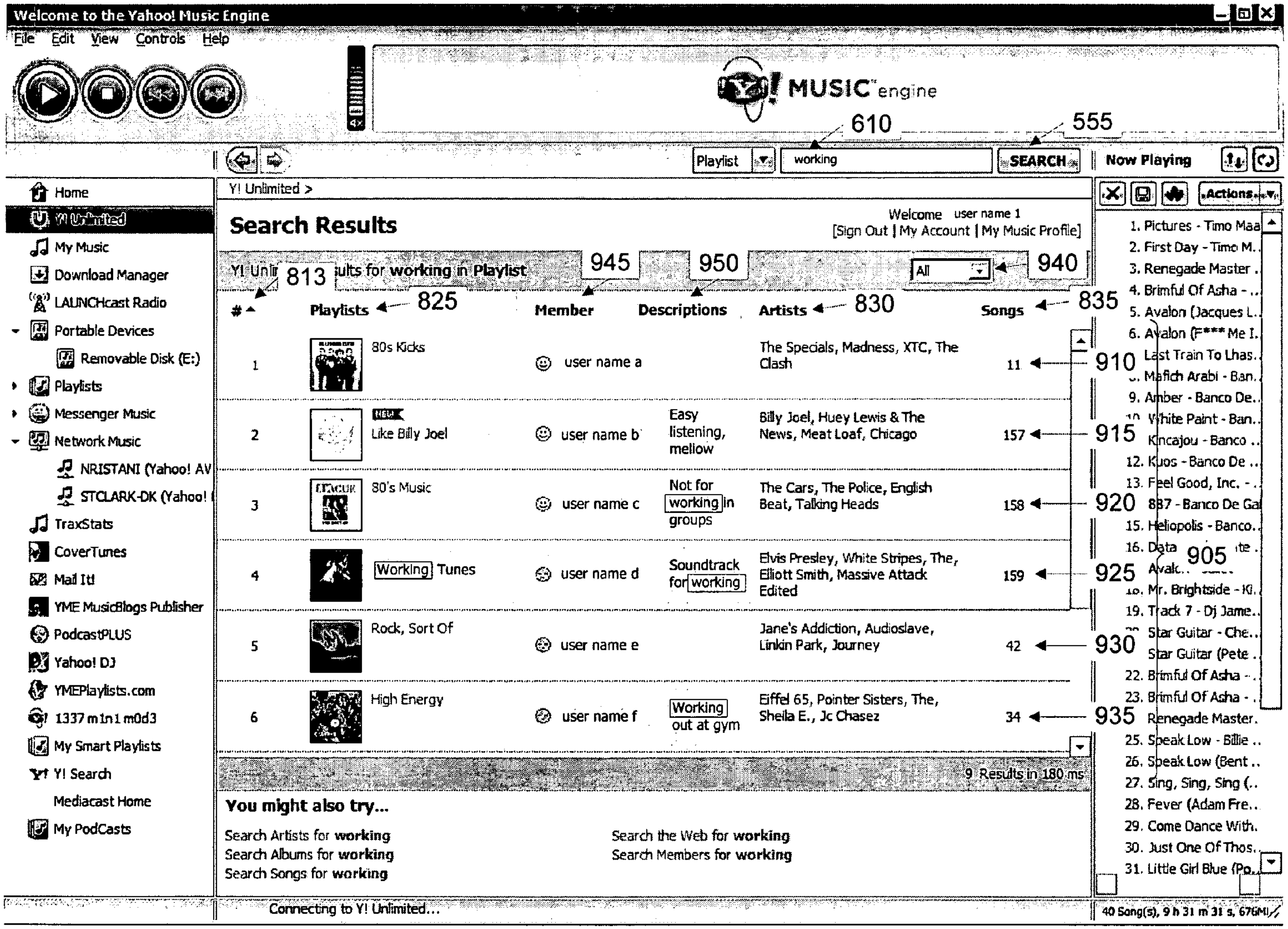
Playlists Display: All

#	Play	Playlists	Artists	Total Songs
1	▶	Rock Songs	The Rolling Stones, U2	5
2	▶	Best of Electronica	Basement Jaxx, Carl Cox, Chemical Brothers, The, Donald Glaude	39
3	▶	Remixes	Chemical Brothers, The, Cornershop, Dj James Diamond, Killers, The	13
4	▶	Bring on the House	Adam Freeland, Deep Dish, The Chemical Brothers, Timo Maas	23
5	▶	Modern Rock	Beck, Death Cab For Cutie, Jamie Lidell, The Postal Service	12
6	▶	Ska Mix	Common Sense, The Mighty Mighty Bosstones, The Specials, Toasters, The	12
7	▶	Fun Beats	Basement Jaxx, Deep Dish, Fatboy Slim, Fluke	18
8	▶	Essential Mixes	Darren Emerson, Deep Dish, Derrick Carter, Hernan Cattaneo	13

1. Paint It Black - The Rollin...
2. I Still Haven't Found Wh...
3. With Or Without You - U2
... re The Streets Have...
... Jay Bloody Sunday - ...

5 Song(s), 0 h 24 m 03 s, 0MB

FIG. 8



Search Results

Welcome user name 1
[Sign Out | My Account | My Music Profile]

Y! Unli[813]ults for working in Playlist

#	Playlists	Member	Descriptions	Artists	Songs
1	80s Kicks	user name a		The Specials, Madness, XTC, The Clash	11
2	Like Billy Joel	user name b	Easy listening, mellow	Billy Joel, Huey Lewis & The News, Meat Loaf, Chicago	157
3	80's Music	user name c	Not for working in groups	The Cars, The Police, English Beat, Talking Heads	158
4	Working Tunes	user name d	Soundtrack for working	Elvis Presley, White Stripes, The, Elliott Smith, Massive Attack Edited	159
5	Rock, Sort Of	user name e		Jane's Addiction, Audioslave, Linkin Park, Journey	42
6	High Energy	user name f	Working out at gym	Eiffel 65, Pointer Sisters, The, Sheila E., Jc Chasez	34

9 Results in 180 ms

You might also try...

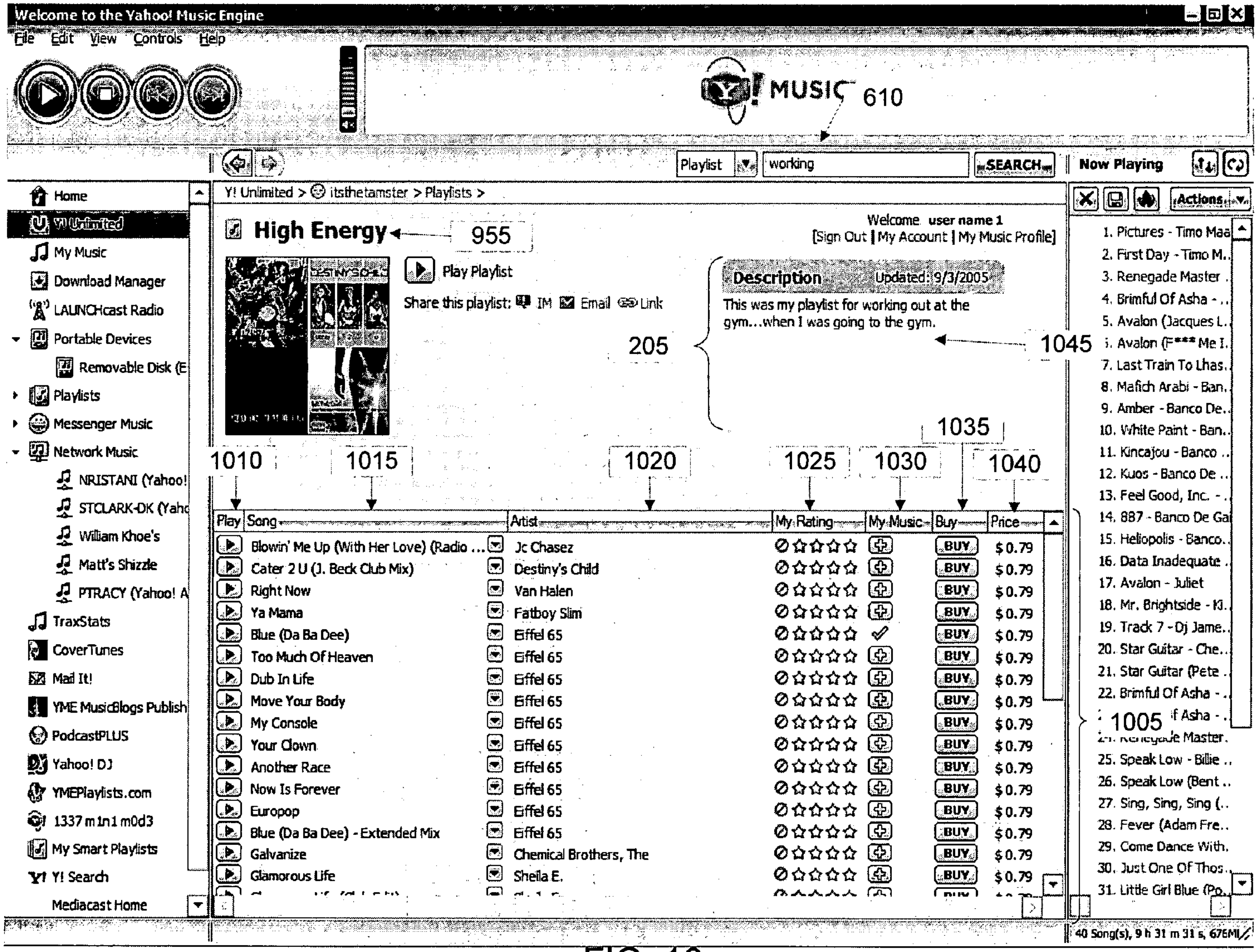
Search Artists for **working**
Search Albums for **working**
Search Songs for **working**
Search the Web for **working**
Search Members for **working**

Connecting to Y! Unlimited...

1. Pictures - Timo Maa
2. First Day - Timo M..
3. Renegade Master .
4. Brimful Of Asha - ..
5. Avalon (Jacques L..
6. Avalon (F*** Me I.
Last Train To Lhas.
Mafich Arabi - Ban.
9. Amber - Banco De..
White Paint - Ban..
Kincajou - Banco ..
12. Kuos - Banco De ..
13. Feel Good, Inc. - ..
887 - Banco De Ga
15. Heliopolis - Banco..
16. Data
Avalo..
Mr. Brightside - Ki.
19. Track 7 - Dj Jame..
Star Guitar - Che..
Star Guitar (Pete .
22. Brimful Of Asha - ..
23. Brimful Of Asha - ..
Renegade Master.
25. Speak Low - Billie ..
26. Speak Low (Bent ..
27. Sing, Sing, Sing (..
28. Fever (Adam Fre..
29. Come Dance With.
30. Just One Of Thos.
31. Little Girl Blue (Po..

40 Song(s), 9 h 31 m 31 s, 676MB

FIG. 9



Welcome to the Yahoo! Music Engine

File Edit View Controls Help

Playlist working SEARCH

Now Playing

Home
Y! Unlimited
My Music
Download Manager
LAUNCHcast Radio
Portable Devices
Removable Disk (E
Playlists
Messenger Music
Network Music
NRISTANI (Yahoo!
STCLARK-DK (Yahc
William Khoe's
Matt's Shizzle
PTRACY (Yahoo! A
TraxStats
CoverTunes
Mail It!
YME MusicBlogs Publish
PodcastPLUS
Yahoo! DJ
YMEPlaylists.com
1337 m1n1 m0d3
My Smart Playlists
Y! Search
Mediacast Home

Y! Unlimited > itsthetamster > Playlists >

High Energy

Play Playlist

Share this playlist: IM Email Link

Welcome user name 1
[Sign Out | My Account | My Music Profile]

Description Updated: 9/3/2005

This was my playlist for working out at the gym...when I was going to the gym.

Play	Song	Artist	My Rating	My Music	Buy	Price
▶	Blowin' Me Up (With Her Love) (Radio ...	Jc Chasez	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Cater 2 U (J. Beck Club Mix)	Destiny's Child	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Right Now	Van Halen	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Ya Mama	Fatboy Slim	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Blue (Da Ba Dee)	Eiffel 65	⊘☆☆☆☆	✓	BUY	$0.79
▶	Too Much Of Heaven	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Dub In Life	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Move Your Body	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	My Console	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Your Clown	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Another Race	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Now Is Forever	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Europop	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Blue (Da Ba Dee) - Extended Mix	Eiffel 65	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Galvanize	Chemical Brothers, The	⊘☆☆☆☆	⊕	BUY	$0.79
▶	Glamorous Life	Sheila E.	⊘☆☆☆☆	⊕	BUY	$0.79

Actions

1. Pictures - Timo Maa
2. First Day - Timo M..
3. Renegade Master .
4. Brimful Of Asha - ..
5. Avalon (Jacques L..
6. Avalon (F*** Me I.
7. Last Train To Lhas.
8. Mafich Arabi - Ban.
9. Amber - Banco De..
10. White Paint - Ban..
11. Kincajou - Banco ..
12. Kuos - Banco De ..
13. Feel Good, Inc. - .
14. 887 - Banco De Ga
15. Heliopolis - Banco..
16. Data Inadequate .
17. Avalon - Juliet
18. Mr. Brightside - Ki.
19. Track 7 - Dj Jame..
20. Star Guitar - Che..
21. Star Guitar (Pete .
22. Brimful Of Asha - ..
23. ...f Asha - .
24. Renegade Master.
25. Speak Low - Billie ..
26. Speak Low (Bent ..
27. Sing, Sing, Sing (..
28. Fever (Adam Fre..
29. Come Dance With.
30. Just One Of Thos..
31. Little Girl Blue (Po..

40 Song(s), 9 h 31 m 31 s, 676MB

FIG. 10

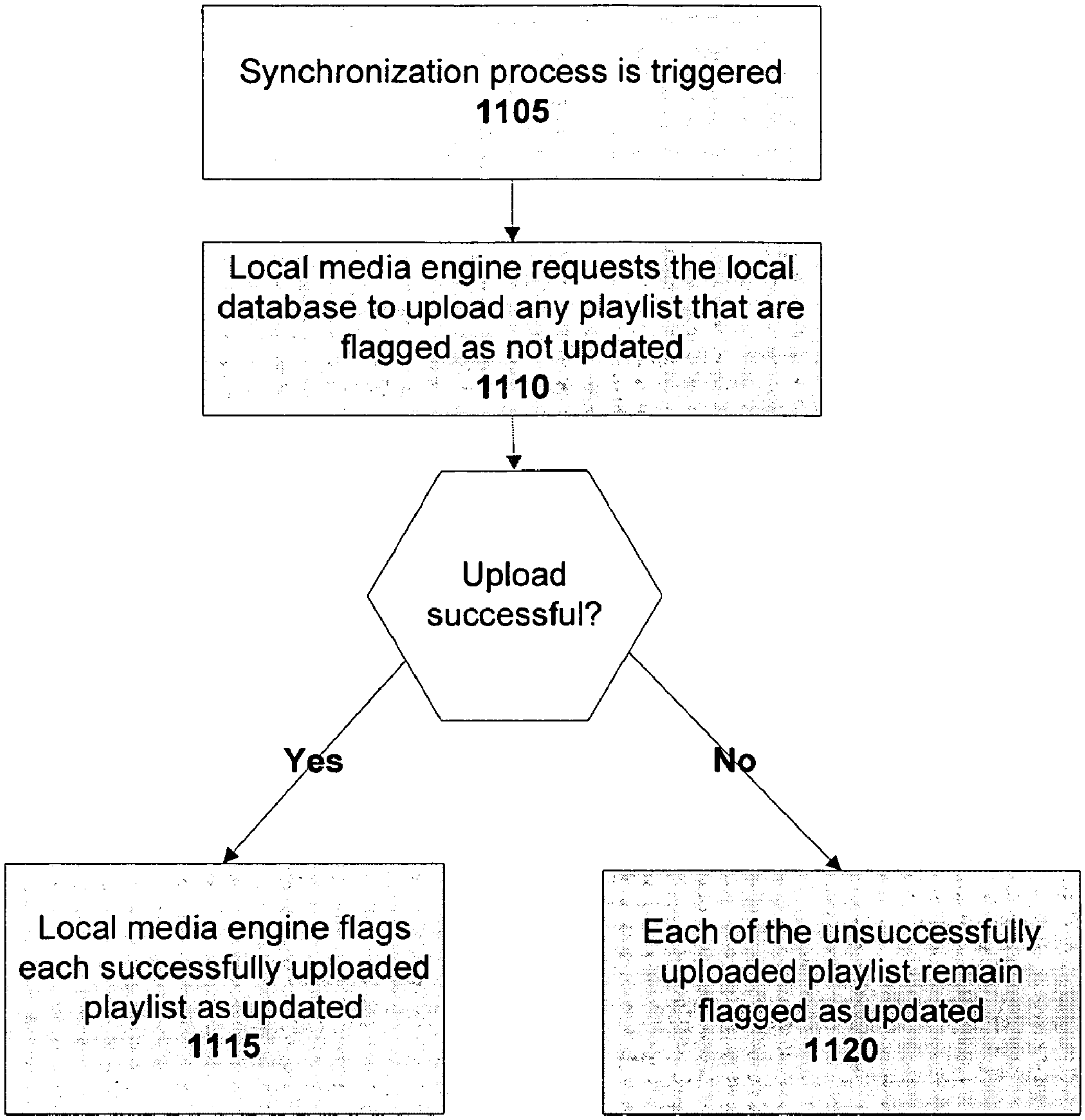

Synchronization process is triggered
1105
Local media engine requests the local database to upload any playlist that are flagged as not updated
1110
Upload successful?
Yes
No
Local media engine flags each successfully uploaded playlist as updated
1115
Each of the unsuccessfully uploaded playlist remain flagged as updated
1120

FIG. 11

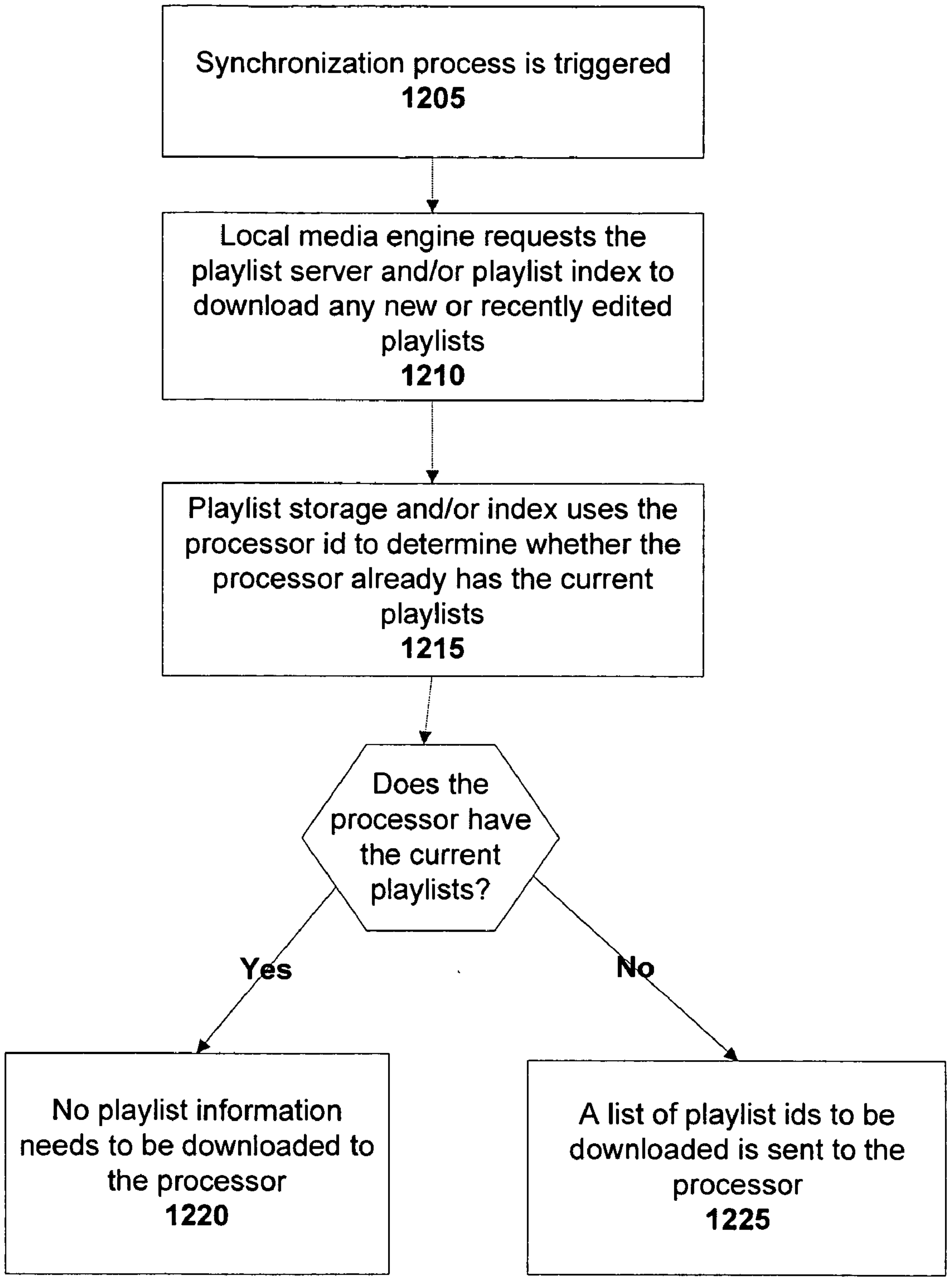

Synchronization process is triggered
1205
Local media engine requests the playlist server and/or playlist index to download any new or recently edited playlists
1210
Playlist storage and/or index uses the processor id to determine whether the processor already has the current playlists
1215
Does the processor have the current playlists?
Yes
No
No playlist information needs to be downloaded to the processor
1220
A list of playlist ids to be downloaded is sent to the processor
1225

FIG. 12

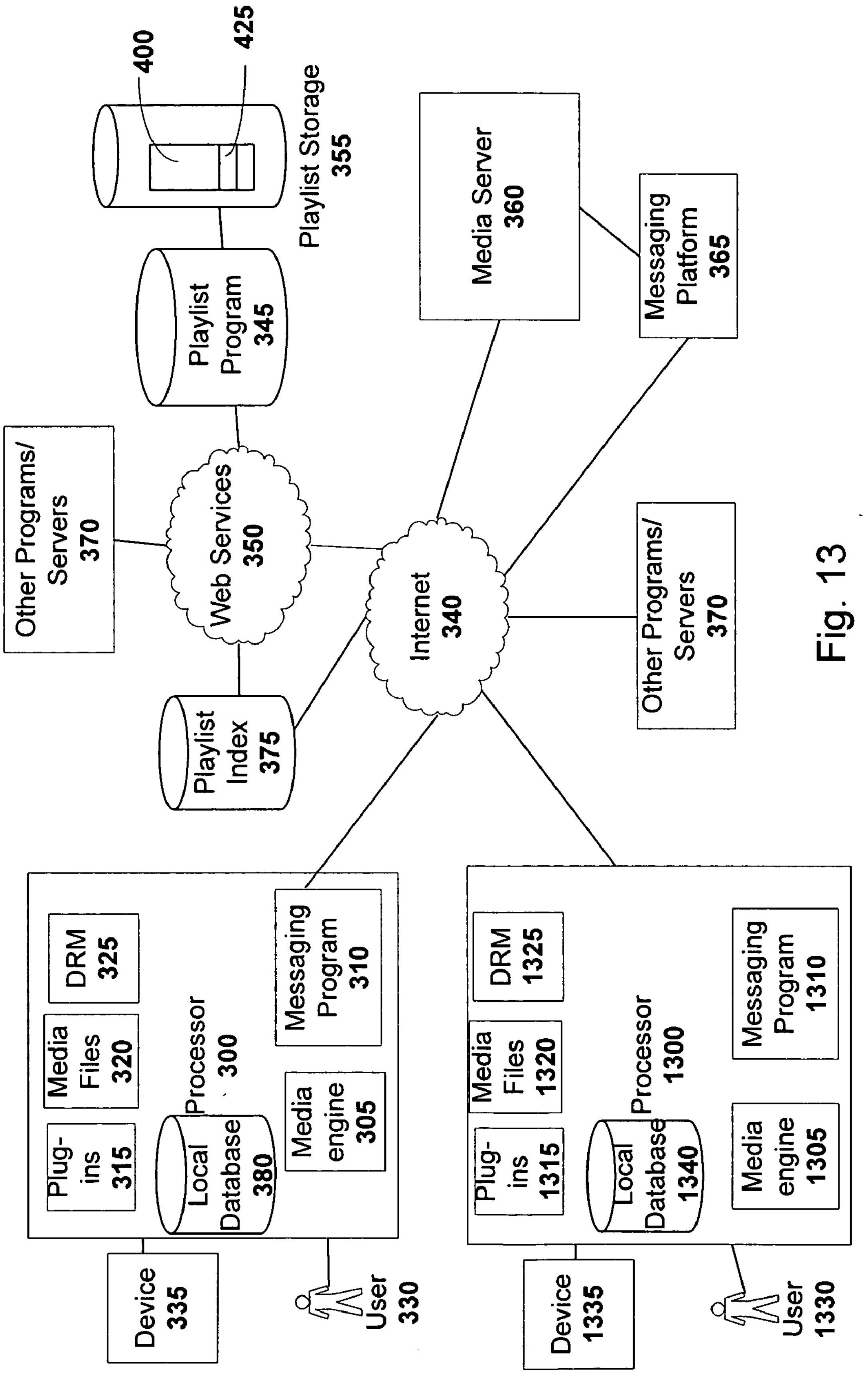

400
425
Playlist Storage
355
Playlist
Program
345
Media Server
360
Messaging
Platform
365
Other Programs/
Servers
370
Web Services
350
Internet
340
Other Programs/
Servers
370
Playlist
Index
375
Plug-
ins
315
Media
Files
320
DRM
325
Processor
300
Local
Database
380
Media
engine
305
Messaging
Program
310
Device
335
User
330
Plug-
ins
1315
Media
Files
1320
DRM
1325
Processor
1300
Local
Database
1340
Media
engine
1305
Messaging
Program
1310
Device
1335
User
1330

Fig. 13

METHOD FOR SHARING AND SEARCHING PLAYLISTS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application Ser. No. 60/657,222, filed Feb. 28, 2005, entitled A SYSTEM AND METHOD FOR DELIVERING MEDIA OVER A NETWORK, and U.S. Provisional Application Ser. No. 60/678,718, filed May 5, 2005, entitled A SYSTEM AND METHOD FOR DELIVERING MEDIA OVER A NETWORK, both of which are hereby incorporated herein by reference.

BACKGROUND

[0002] The expansion of the Internet and the World Wide Web ("web") has given computer users the enhanced ability to listen to and to watch various different forms of media through their computers. This media can be in the form of audio music, music videos, television programs, sporting events or any other form of audio or video media that a user wishes to watch or listen to. As time evolved, users were able to take the music that was on that compact disc store it on their computers and listen to it locally. More recently, web systems were developed which allowed a user log on to a network such as the Internet and, using a media player, either listen to specific media chosen from a group of media that a particular service has to offer or listen to an array of media offerings such as a radio station wherein different songs or other forms of media are combined to allow a user to listen to a group of songs in sequence, whether chosen by the user or by the operators of the network service. For example such a service is Yahoo!™ Music. Many media players, services and other software tools enable media to be organized by a user or a service into playlists, which, as the name connotes, are groups or lists of media files or file identifiers that can be used to effect playback by a user in a sequence or other selected or random order. Further, once a playlist is created, a user may manipulate and edit the playlists freely. However, what is lacking in each of the currently available systems is a method in which a user may easily share or make publicly available created playlists wherein such public playlists are searchable by key input parameters. Further, the current art lacks a sophisticated method of creating a playlist searchable by user-defined descriptor information.

SUMMARY OF THE DISCLOSURE

[0003] In one embodiment, the present disclosure provides a method for making a playlist available to the public. In another embodiment, the present disclosure provides a method for creating a playlist comprising user-defined descriptor information, which can be entered as free form text or prose rather than preselected information like a static genre or short title. In another embodiment, the present disclosure provides a method for searching for such public playlists.

[0004] In one embodiment, the present disclosure provides a method of identifying a published playlist comprising receiving, from a searching entity, search criteria representing a desire to identify a playlist; accessing a playlist storage, the playlist storage comprising user published playlists, some or all of the published playlists comprising associated user-defined playlist descriptor information, the user-defined playlist descriptor information comprising user developed text; searching the playlist storage to identify a desired playlist result set comprising user published playlists with user-defined playlist descriptor information relevant to the search criteria; and communicating the desired playlist result set to the searching entity.

[0005] In one embodiment, the searching entity comprises a user located on a computing device. In one embodiment, the searching entity comprises a software program.

[0006] In one embodiment, the search criteria comprises a mood associated with the desired playlist, a situation associated with the desired playlist, and/or an event associated with the desired playlist.

[0007] In one embodiment, the playlist storage is located on a server, on the internet, and/or is located as a part of an online media service.

[0008] In one embodiment, the desired playlist result set comprises a playlist identifier. In one embodiment, the identifier is a uniform resource locator, a pointer, and/or a universal resource indicator.

[0009] In one embodiment, the desired playlist result set comprises metadata associated with the desired playlist results set, and/or a permission level for accessing the desired playlist results. In one embodiment, the permission level is associated with an online media service. In one embodiment, the desired playlist result set comprises an id associated with the desired playlist.

[0010] In one embodiment, the method of identifying a published playlist further comprises facilitating the display of the desired playlist result set at computing device associated with the searching entity.

[0011] In one embodiment, the present disclosure provides for a method of creating a searchable playlist comprising: receiving at least one media file selection indicating the desire to include at least one media file in a playlist; receiving user-defined playlist descriptor information associated with the playlist, the user-defined playlist descriptor information comprising user developed text; receiving an input indicating the desire to designate the playlist as public, the public designation causing user-defined playlist descriptor information and information associated with the playlist to be searchable by a searching entity; and communicating to a playlist storage, user-defined playlist descriptor information and the associated playlist information so that the user developed text can be searched by a searching entity.

[0012] In one embodiment, identifying a published playlist, the playlist storage is located on a server, and/or the internet.

[0013] In one embodiment, the playlist storage is located as a part of an online media service.

[0014] In one embodiment, the published playlist information comprises information associated with at least one media file, the title associated with the media file, the artist associated with the media file, the genre associated with the media file, and/or a rating associated with the media file.

[0015] In one embodiment, the searching entity comprises a user and/or a software program.

[0016] In one embodiment, the user developed text is searchable by search input criteria. In one embodiment, the search criteria comprises a mood associated with the playlist, a situation associated with the desired playlist, and/or an event associated with the desired playlist.

[0017] In one embodiment, the user defined text comprises a mood associated with the desired playlist, a situation associated with the desired playlist, and/or an event associated with the desired playlist.

[0018] In one embodiment, the format comprises a set of searchable field entries. In one embodiment, the set of searchable field entries comprise a portion of the user defined text, all of the user defined text, and/or a portion of the playlist information.

[0019] In one embodiment, the method of creating a searchable playlist further comprises synchronizing, upon receiving an edit to the playlist, the user-defined playlist descriptor information and the associated playlist information with the playlist storage.

[0020] In one embodiment, the edit comprises an input indicating the desire to designate the playlist as private.

[0021] In one embodiment, storing the user-defined playlist descriptor information and the associated playlist information into a format comprises serializing the user-defined playlist descriptor information and the associated playlist information.

[0022] In one embodiment, the method further comprises synchronizing, upon receiving an edit to the playlist, the user-defined playlist descriptor information and the associated playlist information with the playlist storage.

[0023] In one embodiment, the present disclosure provides one or more computer-readable media having computer-readable instructions thereon which, when executed by one or more computing devices, implements the method of creating a searchable playlist comprising: receiving at least one media file selection indicating the desire to include at least one media file in a playlist; receiving user-defined playlist descriptor information associated with the playlist, the user-defined playlist descriptor information comprising user developed text; receiving an input indicating the desire to designate the playlist as public, the public designation causing user-defined playlist descriptor information and information associated with the playlist to be searchable by a searching entity; and communicating to a playlist storage, user-defined playlist descriptor information and the associated playlist information so that the user developed text can be searched by a searching entity.

[0024] In one embodiment, the present disclosure provides one or more computer-readable media having computer-readable instructions thereon which, when executed by one or more computing devices, implements the method of identifying a published playlist comprising receiving, from a searching entity, search criteria representing a desire to identify a playlist; accessing a playlist storage, the playlist storage comprising user published playlists, some or all of the published playlists comprising associated user-defined playlist descriptor information, the user-defined playlist descriptor information comprising user developed text; searching the playlist storage to identify a desired playlist result set comprising user published playlists with user-defined playlist descriptor information relevant to the search criteria; and communicating the desired playlist result set to the searching entity.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] The following drawing figures, which form a part of this application, are illustrative of embodiments of the present disclosure and are not meant to limit the scope of the disclosure in any manner, which scope shall be based on the claims appended hereto.

[0026] **FIG. 1** is an exemplary flowchart illustrating a method according to one exemplary embodiment;

[0027] **FIG. 2** is an exemplary user interface according to one exemplary embodiment;

[0028] **FIG. 3** is schematic illustrating an exemplary system architecture of one embodiment;

[0029] **FIG. 4** is a schematic representing a playlist entry according to an exemplary embodiment;

[0030] **FIG. 5** is an exemplary user interface according to one exemplary embodiment;

[0031] **FIG. 6** is an exemplary user interface according to one exemplary embodiment;

[0032] **FIG. 7** is an exemplary user interface according to one exemplary embodiment;

[0033] **FIG. 8** is an exemplary user interface according to one exemplary embodiment;

[0034] **FIG. 9** is an exemplary user interface according to one exemplary embodiment;

[0035] **FIG. 10** is an exemplary user interface according to one exemplary embodiment;

[0036] **FIG. 11** is an exemplary flowchart illustrating a method according to one exemplary embodiment;

[0037] **FIG. 12** is an exemplary flowchart illustrating a method according to one exemplary embodiment; and

[0038] **FIG. 13** is a schematic illustrating an exemplary system architecture of one embodiment.

DETAILED DESCRIPTION

[0039] In one embodiment, the present disclosure provides a method for making a playlist available to the public. In another embodiment, the present disclosure provides a method for creating a playlist comprising user-defined descriptor information, which can be entered as free form text or prose rather than pre-selected information like a static genre or short title. In another embodiment, the present disclosure provides a method for searching for such public playlists.

[0040] In one embodiment the ability to publish, create, and search for a playlist is facilitated by a media management program. In a further embodiment a user interface of such a media management program allows a user to publish and search for playlists. In other embodiments, non-media management applications, programs, and/or servers are capable of searching for such published playlists.

[0041] **FIG. 3** depicts an exemplary system configuration supporting the ability to publish, create, and search for a playlist. As can be seen in **FIG. 3**, the exemplary system configuration comprises a device **335** connected to a processor **300** associated with a user **330**. Further embodiments contemplate multiple users each connected to the system configuration via a processor as depicted in **FIG. 13**. In one embodiment, the device may comprise any type of computing device capable of rendering media files. In another embodiment, the processor **300** may be located on a computing device, which can be, by way of non-limiting example, a personal computer, cellular telephone, personal digital assistant, or a portable media player, or any other device now known or to become known capable of carrying out the features and functions described herein. As seen in **FIG. 3**, the processor may comprise a media engine **305**, a messaging program **310**, digital rights management (drm) **325**, media files **320**, plug-in applications **315**, and a local database **380** for storing information such as playlist information. In addition, through the internet **340**, the processor **300** is connected to a media server **360**, other programs and servers **370** and a messaging platform **365**. Of course, the processor can comprise any combination of necessary components know to those skilled in the art in order to perform the functions described herein. Further depicted in **FIG. 3**, the processor **300** is connected via the internet **340** to a playlist program **345**, a playlist storage **355**, a playlist index **375**, webservices **350**, a media server **360**, messenger platform **365**, and other programs and servers **370**. In one embodiment, the playlist program **345** manages the exchange of information to and from the playlist storage **355**. In one embodiment, the playlist storage **355** stores all related playlist information. For example, as depicted in **FIG. 3**, the playlist storage stores playlist entries comprising user-defined descriptor information as further described herein. In one embodiment, the playlist index **375** stores a subset of the playlist information stored in the playlist storage **355** so that the processor **300** and/or other programs and servers **370** can easily and efficiently access the stored playlist information. In a further embodiment, only playlists which have been designated as public are stored in the playlist index **375**. Thus, the playlist index **375** may be accessed when searching for public playlists. In one embodiment, the web services **350** provide efficient manner or way to access the playlist storage **355** and/or playlist index **375**. In other embodiments, the webservices **350** may not be necessary to access the playlist storage **355** and/or a playlist index **375**. In one embodiment, as depicted in **FIG. 3**, other programs or servers **370** are capable of being connected to the web services **350** and/or internet **340**. For example, in some embodiments, other programs or servers **370** can communicate with the playlist index **375**, playlist program **345**, and/or playlist storage **355** alone or independent of the processor **300** and/or media server **360**. Thus, as depicted in **FIG. 3**, playlists are searchable by any user, program, server, or any entity with an internet connection. In one aspect, the system configuration allows any other program and/or server **370**, such as any non-media management program, to search for playlists and/or store created and/or published playlists for searching. For example, a user could search for playlists, i.e./e. access the playlist index and/or playlist storage, from a search tool found in a downloadable toolbar such as for example the downloadable Yahoo! Toolbar. With reference to **FIG. 3**, multiple variations and combinations of connections among the various components are contemplated. Furthermore, **FIG. 3** is merely exemplary and not meant to limit the system configuration or architecture supporting the present disclosure in any way. Obvious variations of the components and configurations know to those skilled in the art are contemplated.

[0042] The publishing of a playlist according to one embodiment of the present disclosure will now be discussed with reference to **FIG. 1** that depicts an exemplary method of the present disclosure. In one embodiment, referring to **FIG. 1**, first, the user creates **105** a playlist. This playlist can be created in any number of different ways (i.e., user created, criteria playlists, affinity playlists) as known in the art. In one embodiment, a playlist comprises at least one media file and/or media file identifier. In one embodiment, once the playlist is created or during playlist creation, a user is able to enter user-defined descriptor information with respect to the playlist. Unlike current published playlists available in the art, user-defined descriptor information permits a user to creatively enter free-form text, prose, tags or keywords that the user believes describe characteristics of or feelings or moods set by the user-published playlist. For example, present published playlists permit a user to identify a title, genre, and/or artist information for the playlist. This information is not entirely useful to a third party seeking to find a playlist for a specific purpose, environment, mood, concept, or task, for example. User-defined descriptor information of the present disclosure, as used herein, refers to any type of user-provided text, prose, tags, keywords and/or descriptors describing a user's understanding, perception and/or feelings toward, or the mood set by, a playlist, as opposed to the less descriptive associated metadata namely playlist title, or artist, genre, or media file title, which are static and while factual may not satisfactorily provide sufficient characteristic information. One exemplary embodiment of a user interface capable of receiving user-defined descriptor information is depicted in **FIG. 2**. Referring to **FIG. 2**, the user interface comprises a text field box **205** in which a user can enter user-defined descriptor information. In one embodiment, the user-defined descriptor information may comprise a playlist description as entered by the user. Then, by selecting a user input indicia, such as a button **230**, the user can save the user-defined playlist information along with the playlist. For example, **FIG. 5** depicts a text description box **205** comprising user-defined descriptor information. In **FIG. 5**, the playlist description box **205** includes the user-defined descriptor information "playlist for while at work. Not ideal for working out." In one embodiment, each time the playlist is accessed, the user-defined descriptor information also appears along with the playlist. In further embodiments, the user can edit the user-defined descriptor information as many times as the user desires. In another embodiment, the user-defined playlist information entered into the descriptive field box **205** is indexed or serialized along with the playlist information into a storable playlist entry as described in further detail herein.

[0043] Referring back to **FIG. 1** once the playlist is created **105**, a playlist entry is created **110**. In one embodiment, the playlist entry is created **105** by serializing the playlist information and user-defined descriptor information into a searchable playlist entry format. In one embodiment, playlist associated metadata may comprise metadata about the playlist itself such as the playlist name, date created, the duration of the playlist etc. Further, the playlist associated metadata may comprise metadata about each media file within the playlist such as the title of the media file, artist, and album, length, date of creation, and/or rating. In one embodiment, the searchable playlist entry format may comprise a series of searchable fields. For instance, an xml file or XSPF file structure. For example, XSPF standard comprises elements or fields such as title, annotation or comment, identity of the creator, info such as a uri or url, canonical ID, image associated with the playlist, creation date, any licensing term, etc. Thus, in one embodiment, when a playlist is created **105**, the playlist information and user-defined descriptor information is serialized into a searchable format.

[0044] **FIG. 4** depicts an exemplary playlist entry **400** derived from a playlist as depicted in **FIG. 5**. Referring to **FIG. 5**, the exemplary playlist **505** entitled "Example Playlist" comprises five media files each listed by song title **510**, artist name **515**, an indicator **520** of whether the playlist is burnable to a cd or not, the genre **525** associated with the media file, the duration **530** of the particular media file, and a rating **535** associated with a media file. For example, the first media file **540** in the playlist **505** is entitled "Brick," by the artist "Ben Folds Five" which has duration of four minutes and forty-eight seconds. Further, as can be seen in the user interface, the media file **540** is not burnable and is in the Modern Rock genre. Additionally, the playlist description text box **505** has been entered with user-defined descriptor information **505** corresponding to the playlist. Accordingly, the playlist entry corresponding to the playlist **505** can comprise some, all, or none of the information displayed in the user interface. Of course other, non-displayed information associated with the media files within the playlist may comprise the playlist entry.

[0045] Accordingly, **FIG. 4** depicts an exemplary playlist entry corresponding to the playlist as depicted **FIG. 5**. As can be seen in **FIG. 4**, the exemplary playlist entry **400** comprises several fields **405-435**. As depicted in **FIG. 4**, one field **405** comprises a listing of all the titles associated with the media files in the playlist. As seen, field **405** lists: Brick, Amber, Grace, It Ends Tonight, and Only the Good Die Young. Accordingly, with reference to **FIG. 5**, these are the same titles associated with the media files of the playlist **505**. Similarly, another field **410** comprises a listing of all the artists associated with the media files in the playlist. As seen, field **405** lists: Ben Folds, **311**, Jeff Buckley, All-American Rejects, and Billy Joel. Accordingly, with reference to **FIG. 5**, these are the same artists associated with the media files of the playlist **505**. Further, field **420** lists the name of the playlist creator, field **420** lists the name of the playlist, and field **430** lists the genre associated with each media file in the playlist. In further embodiments, the playlist entry comprises a field **425** comprising strings of text representing the user-defined descriptor information into the field description box **205** as depicted in **FIG. 5**. For example, referring to **FIG. 5**, the text field description box **205** contains the user-defined playlist information: "Playlist for while at work" and "Not ideal for working out." As described herein, in one embodiment, the playlist application is capable of serializing the user-defined descriptor information entered into the box into a useful field of the playlist entry. In one embodiment, the field **425** may comprise some or all of the text entered into the text field description box **205**. Further, all methods known to those skilled in the art of serializing text into a searchable field are contemplated. Thus, the playlist is searchable by user-defined descriptor information. As can be seen in **FIG. 4**, field **425** comprises the text strings "Playlist for while at work" and "Not ideal for working out." Further depicted in **FIG. 4**, a field entry **435** may comprise pointers to the contents of the playlist and/or physically include the contents of the playlist. In a further embodiment a playlist field entry may comprise a playlist id assigned and recognized among an online media service. Accordingly, in one embodiment, a user or entity is capable of searching the contents of the playlist entry. Thus, the user-defined descriptor information can be used to further enhance indexing and searching capabilities. It is noted that not all of the fields in the playlist entry may be capable of searching. In further embodiments, it may be desirable to indicate which types of fields are not preferred for searching such as the member's user name or profile name. Again, in one embodiment, the playlist entry can comprise any combination of field entries gathered from different sources. Thus, in one embodiment, the categorical aggregation of fields into a useful playlist entry allows for efficient playlist searching capabilities.

[0046] Referring back to **FIG. 1**, once the playlist is created **105** and the playlist entry is created **110**, the playlist entry is stored **115** in a playlist index and stored **120** in a playlist storage. In one embodiment, with reference to **FIG. 3**, the playlist storage **345** stores the playlist entries. Thus, the playlist storage contains all playlist information. In one embodiment, the playlist index **375** stores a subset of the playlist entries stored in the playlist storage **355** so that the processor **300** and/or other programs and servers **370** can easily and efficiently access the stored playlist entries.

[0047] Turning back to **FIG. 1**, in one embodiment, the user can elect to publish **125** a playlist. In one embodiment, when the playlist is selected as published **125**, the playlist entry is updated within the playlist index **130** and within the playlist storage **135**. Similarly, when a public playlist is elected to become private (not depicted), the playlist entry is updated within the playlist index **130** and within the playlist storage **135**. In one embodiment, the playlist index **130** and/or playlist storage **135** are not updated simultaneously upon publication or privatization. Instead, the playlist index **130** and/or playlist storage **135** are updated upon a predetermined time interval. In one embodiment, only public playlists are stored in the playlist index. Thus, by accessing the playlist index for searching a playlist, only published playlists are searchable.

[0048] Accordingly, in one embodiment, a user's playlist is by default considered private. Thus, a user must affirmatively choose to make a playlist public. In one embodiment, a user may choose to publish their playlist, for example, by clicking on a user input indicia such as a button or link. The user is preferably given the option of making the published playlist public, thus allowing other users to not only view the playlist but to listen to media listed in the playlist as well. In one embodiment, the published playlist can contain pointers that identify the network location(s) at which the media file resides or may be accessed (a pointer may comprise a URL or URI), metadata or actual media files, although preferably the playlist contains pointers to avoid the need to store multiple copies of the same media file on the network. Thus, a playlist can be made available to a user at any location by publishing the playlist. Thus, the ability to publish a playlist has the added benefit of making the playlist completely portable by being available to all users, as it will be residing on the media server or other user accessible location(s) on the network.

[0049] For example, referring to **FIG. 2**, through an exemplary user interface, the user is able to select the public feature **210** that publishes the playlist. In one embodiment, selecting the public feature **210** causes a visual indictor to appear or highlighting or any other way of emphasizing and indicating that a particular playlist is public. For example, in **FIG. 2**, the text of the public feature **210**"Make it public" becomes darken and bolded upon selection. Further, the user can select the private **235** feature at any point to remove the playlist from the public. Similar to selecting the public feature **210**, selecting the private feature may cause a visual indictor to appear or highlighting or any other way of emphasizing and indicating that a particular playlist is private. Further, as depicted in **FIG. 2**, a user can also email **220** a playlist, send **215** the playlist via an instant messaging application, or copy **225** the playlist as a url to share with other users, for example by posting it on a blog or website. In a further embodiment, once a playlist is selected as public, it is published on a profile page which may or may not be incorporated into a media management program as described in further detail herein.

[0050] In one embodiment, the playlists are searchable through a user interface as depicted in **FIG. 5**. As can be seen in **FIG. 5**, the user interface comprises an area **560** for searching for playlists. In one exemplary embodiment, the area for searching **560** for playlists comprises a drop down menu **545** and a text field box **550**. As can been seen in **FIG. 5**, the user can select the "playlist" choice from the drop down menu **545**, enter any search criteria into the text field box **550**, and select search indicia such as a button **555** to trigger a search for at least one playlist associated with the search criteria. In one embodiment, the playlist program **345** as depicted in **FIG. 3** can receive input criteria from a user **330** or other programs and services **370**, reference the playlist entries and retrieve and return the appropriate results back to the user's processor **300** and/or other programs and servers **370**. In one embodiment, the processor **300** and/or other programs and servers **370** can directly search the playlist index without use of the playlist program. In one embodiment, referring to **FIG. 3**, when a search input for a playlist is received from the processor **300** and/or other programs and servers **370**, it queries the playlist index **375** for matching playlists. In one embodiment, some or all of the information contained in a playlist entry is searchable. In other embodiments, playlist entry information may be included in the entry and used for display in the user interface, but may not be searchable. This could include the number of songs or media files in the playlist, the images associated with the songs in the playlist, the date the playlist was created and modified, and the genre names of the songs in the playlist. Of course, any other associated playlist information may be used as a field entry and may or may not be searchable.

[0051] In one embodiment, the playlist search results are returned via a user interface which displays the resultant playlists. In one embodiment, the user interface may be incorporated in a media management program. For example, in **FIG. 6**, a user can search for playlists by choosing the playlist parameter **605** from a drop down menu **545** and entering search criteria **610** into a text field box **550**. Then, a user can select a button **555** to trigger a search for playlists associated with the search criteria.

[0052] In one embodiment, the search criteria are used look for matches against the content of all the playlist field entries. For example, if the search criterion **610** is for the term "working" as depicted in **FIG. 6**, each playlist entry is referenced looking for the term "working." For example, the term "working" may relate to playlists for a working environment or even return playlists for working out. In any event, the input criterion can search by any string of text representing any portion or entire concept, situation, event, and/or mood. In one embodiment, if any playlist field contains the term, the playlist will be returned to the searching entity. For example, a playlist will be returned if the term "working" is found in the playlist title field, user-defined descriptor information field, artist field, playlist title field, and/or genre field etc. It is important to note that in some embodiments, the playlists storage and playlist index are capable of storing playlist entries from multiple users and multiple sources. Thus, when a playlist search is triggered, a user or entity can search playlists created by any other user or source. Thus, the present disclosure allows users or entities to search for any playlist indexed and stored in the playlist storage and/or playlist index from any creator and from any source.

[0053] An exemplary embodiment of the returned results from a playlist search is depicted in **FIG. 9**. As can be seen in **FIG. 9**, in one embodiment, the playlists matching the search criteria are displayed in a scrollable area **905**. As depicted in **FIG. 9**, the results of the playlist search may be displayed in columnar format. In other embodiments, the results of the playlist search may be displayed in table format, in a menu, drop down list or any other suitable way. Further, area **905** comprises a list of each of the matching playlist **910**, **915**, **920**, **925**, **930**, and **935**. In one embodiment, as depicted in **FIG. 9**, each playlist is numbered **815**, listed by title **825**, by the member and/or user **945** who created and/or owns the playlist, key artists **830**, and number **835** of media files within the playlist. In a further embodiment, as depicted in **FIG. 9**, a playlist may also have listed a description column **950**. In one embodiment, the playlist description **950** column lists an excerpt of terms from the user-defined descriptor information text box associated with each playlist. For example, as shown in **FIG. 9**, the second returned playlist **915** has descriptive terms such as "easy listening" and "mellow" and the third returned playlist **920** has description terms such as "not for working in groups." Further, in cases where no user-defined descriptor information has been entered and associated with the returned playlist, those returned playlists will not display a description. For example, the first returned playlist **910** has a blank space in the description column **950**. In other embodiments, the description column **950** may comprise a URL, uri, or a pointer that directs a user to the full user-defined descriptor information. In other embodiments, simply toggling over the terms in the description column **950** will cause the full description to appear as an enlarged text box.

[0054] In a further embodiment, a user can filter the returned playlist results. For example, as depicted in **FIG. 9**, a drop down menu **940** filters the displayed results by displaying all of the results or some select portion of the displayed playlists. For example, the playlist results may be filtered according to popularity, recency, the identity of the associated members to show only friends' playlists, by usage rights (for example, to display playlists with only subscription content), and/or any identifiable associated playlist attribute such as artist, album or genre. In other embodiments, the filter parameters may be set through a variety of presets or through a preference menu. For example, a preset could be selected for only displaying returned playlists that have the artist matching the search criteria. Of course, many combinations and variations of filter parameters as known to those skilled in the art are contemplated.

[0055] In some embodiments, the search criteria used to identify a playlist is highlighted or otherwise emphasized in the user interface. The search term may be highlighted or emphasized using a particular color, a box around the text, via an icon, or any other known method of highlighting or emphasizing text and/or graphics in a user interface. In one aspect, this allows a user to easily view which field or playlist attribute matched the search criteria. For example, referring **FIG. 9**, since the search criteria **610** was the term "working"; the term is boxed wherever it appears in the returned playlist user interface. As depicted in **FIG. 9**, the term "working" appears in the description column **950** of the third playlist "80's Music"**920**, the fourth playlist "Working Tunes"**925**, and the sixth playlist "High Energy"**935**. Meaning, in one embodiment, each of these playlists has associated user-defined descriptor information comprising the term "working." Also, the term is boxed in the title of the fourth playlist "Working Tunes"**925**. Thus, in one embodiment, the user interface displays the search term for quick and easy reference.

[0056] In other embodiments, instead of highlighting or otherwise emphasizing the search term in the returned playlist display, the display may comprise a column or some type of indicator that indicates where the search term is located with respect to the returned playlist. For example, the indicator or column can indicate that the search term appears in the title of a media file within the playlist, or is a part of the user-defined descriptor information, and so on.

[0057] In one embodiment, the returned playlist results are displayed as media file identifiers such as uris, uris, or pointers that can direct a user to the location of the playlist and/or its contents. Thus, through the user interface as depicted in **FIG. 9**, a user can select any of the returned playlists and/or the member name to further explore information associated with the playlist and/or member such as viewing and/or playing the playlist.

[0058] **FIG. 10** depicts an exemplary user interface if the "High Energy" playlist **935** is selected from the returned playlist user interface depicted in **FIG. 9**. As can be seen in **FIG. 10**, the media files of the selected playlist **935** are listed in an area **1005**. Area **1005** lists each media file in the playlist **935** by media file title **1015**, artist **1020**, and rating **1025**. In a further embodiment, as depicted in **FIG. 10**, a button **1030** allowing a user to add the media file to their local media file collection, a button **1010** allowing a user to play the media file, and a button **1035** allowing a user to purchase the media file and its associated price **1040** are also listed next to each media file of the playlist **915**. In a further embodiment, as depicted in **FIG. 10**, a description text box **205** corresponding to the selected playlist is displayed. As depicted in **FIG. 10**, the description box **205** comprises user defined descriptor information **1045**"This was my playlist for working out at the gym . . . when I was going to the gym." Thus, it can be seen that the search term "working"**610** appears the in the user defined descriptor information **1045** associated with the "High Energy" playlist **935**.

[0059] In a further embodiment, the playlist program is capable of weighting the fields of a playlist in order to determine which playlists with fields matching the search criteria should be returned to the user or entity. For example, the artist name and media file title fields may be weighted more than the creator name field. Thus, a playlist is more likely to be returned if a search criterion is found in the artist name and/or media file name than if in the creator field. In other embodiments, conditional logic is used to determine whether a particular playlist is returned to a user. In other embodiments, a specific count of occurrences of an input criterion in a playlist entry is required in order for a playlist to be returned to a user. In further embodiments, what playlists are returned is based, in part, on affinity relationships. The playlist program can also use the popularity of a playlist as measured by the amount of time other users spend listening to it to influence relevancy by ranking popular playlists higher than unpopular playlists. Playlist size may also be used as part of relevancy to ensure that users who search get useful playlists returned to them. Of course, many combinations and methods of selecting a playlist using search criteria known to those skilled in the art are contemplated.

[0060] In one embodiment, once a playlist is selected to be public, it is published on a profile page of a media management program. In another embodiment, a user's profile page is accessible outside the context of a media management program. Thus, in one embodiment, the profile page is accessible from any authorized internet source. In one aspect, a profile page comprises a page which collects and displays various information about a particular user. In one embodiment, all of a particular user's published playlists may be displayed on the user's profile page. **FIG. 7** depicts an exemplary embodiment of a user profile page. As can be seen in **FIG. 7**, the profile page comprises several areas **705**, **725**, **730**, **740** for displaying various types of information about the user. For example, area **725** displays user-defined text that describes the user's musical interests and/or preferences. Further, area **730** displays users applied media file ratings organized by genre, album, and song, and photos of the user's highest-rated artists. Also, area **740** displays a user's listening preferences. As depicted in **FIG. 7**, the user can edit each area as desired. In particular, area **705** displays a user's published playlists. As can be seen in area **705**, two playlists **715**, **720** are listed. The profile page may show the playlist titles **750**, **745**, along with information such as key artists (not depicted), the number of songs in the playlist (not depicted) and description information **755**, **760**. This description information **755**, **760** may or may not be derived from the playlist's associated user-defined descriptor information. Further, a link **710** is provided so that all of the user's published playlist may be viewed.

[0061] An exemplary embodiment of a user interface displaying all of a user's published playlists is depicted in **FIG. 8**. As can be seen in **FIG. 8**, a user's published playlists are displayed in a scrollable area **810** and are listed by number **815**, playlist title and a representative playlist image **825**, key artists included in the playlist **830**, and a number **835** of total media files within the particular playlist. For example, the first playlist displayed is entitled "Rock Songs" which has five total songs and "The Rolling Stones" and

"U2" as the key artists. In one embodiment, the key artist **830** represents the more popular artists as defined by a user, an online music community, or via ratings within the particular playlist. In another embodiment, the key artist **830** represents the artists with the highest reoccurrence within the playlist. In a further embodiment, the area **810** may also comprise a column for listing user-defined descriptor information (not depicted). In a further embodiment, as depicted in **FIG. 8**, a corresponding categorical menu area **850** comprises a playlist selection **840** which can be expanded **845** to show all playlists. Thus, area **850** allows for easy browsing of a user's playlists. In one embodiment, area **850** provides a visual indicator (not depicted) indicating whether a playlist is public or not. In one embodiment, the visual indicator may comprise highlighting the public playlists, adding a particular icon next to each public playlist, or subdividing the expanded menu **845** (not depicted) into public playlist and private playlists.

[0062] Further, in one embodiment, selecting the play **820** button next to each playlist will cause the playlist to be loaded and played by a media management program (such as media player like Yahoo! Music Engine or MusicMatch). Through such a media management program, a user can further manage their playlists. In one embodiment, the following variations and features regarding managing and the collaborative playlist is controlled by the media management program. For example, a media management program's user interface may allow the user to stop, rewind, fast-forward, pause, and or/stop the playing of the playlist. In another embodiment, the user can copy the playlist and rearrange and/or delete media files to customize the playlist. It is contemplated that users will also be able to bookmark other users' playlists, enabling them to keep access to a playlist even as the creator of the playlist makes changes to it.

[0063] Providing access of a user's published playlist on a profile page allows users to further get to know other member's and browse their musical tastes. Of course any combination and other areas of information are contemplated. In one embodiment, users can control access to their playlists by making an individual playlist public or private, or by making the entire profile page public or private. Of course it is contemplated that access to a user's playlist may be controlled by further detailed designations. For example, a user can designated that all playlists are only publicly available to a designated list of friends, other users, or specific online community such as a messenger community.

[0064] In one embodiment, the ability to share and view playlists among different users will be dependent on certain Digital Rights Management (DRM). In one embodiment, the playlists created and viewed by another user will contain sufficient DRM information to ensure that the user viewing the playlist has sufficient rights or a specific permission level to permit the user to experience the media contained in the playlist. In one embodiment, this function is controlled by the media management program.

[0065] In one embodiment, the ability of a user viewing and/or playing of another user's published playlist to access a media file and/or media file identifier is governed by a permission level. In one embodiment, the permission level may be triggered upon the user attempting to view the playlist. In another embodiment, the permission level may be triggered when the user attempts to play the playlist. Meaning, that in one embodiment, the user can view all playlists, but permission levels will control whether a user can play the media files within the published playlist. Thus, in one embodiment, the ability to view and/or play a media file and/or media file identifier in a published playlist may be governed by a permission level. In one exemplary embodiment, the permission level is associated with the media file. In another exemplary embodiment, the permission level is controlled by a subscription from a service provider. Rights and permissions can be managed by a DRM application or an associated media management application or media player or combination thereof.

[0066] In an exemplary embodiment, the permission level depends on each user's status as a subscriber or a non-subscriber to an on-demand-streaming media service (ODSMS). For example, if the user who has published a playlist is a subscriber and the user desiring to view and/or play the published playlist is also a subscriber, then the desiring user can have full access to each ODSMS media file identifier on the playlist. In another exemplary embodiment, if the user publishing the playlist is a subscriber to an ODSMS and the user desiring to view and/or play the published playlist is a non-subscriber, then the desiring user may have limited access to the ODSMS media files. For example, the desiring user may be limited to a pre-determined length clip of each ODSMS media file, such as a 30-second clip when the desiring user plays the media file. In another example, the desiring user may be given a pre-determined number of times the user can play each ODSMS media file. After this pre-determined number of times has been reached, the user is then limited to a default, pre-determined length of time for each media file (such as a 30-second clip). For example, the non-subscribing, desiring user may be limited to playing an ODSMS media file at its full-length to three times. Thus, once the user had played the media file and/or media file identifier three times, the user can now only listen to 30 second clips of the media file. This allows a non-subscribing desiring user to listen to ODSMS music otherwise not available to the user, while enticing the user to subscribe to the service. Accordingly, in one embodiment, once the user's pre-determined numbers of full-length plays are used, the user is prompted and has the option to subscribe to the ODSMS.

[0067] Of course, these embodiments are not limited to ODSMS media files, but can apply to any media files which require a specified level of sharing, viewing, and/or playing access. For example, it could be negotiated that certain media files generally not available for full-length sharing could be shared and played between any status (subscribing or non-subscribing) of users for a pre-determined number of times. For example, if there is a particular artist who wants to promote and/or entice users to listen to their media files, the artist or their representatives can allow a particular media file to be able to be viewed and/or played among any type of users. However, the users are limited to playing the media file to a certain number of times.

[0068] In another embodiment, the user can further utilize the collaborative playlist by burning the playlist to a CD or transferring the media files to a portable device. For example, the user can burn the media filed listed in playlist in the media player window to a CD and/or transfer the media files to a portable device by clicking a button and selecting the option from a drop-down menu. In some embodiments, the ability to transfer media files to a portable device or burn media files to a CD may depend on whether the user has specified access to those media files. Particularly, the user may be required to pay for files not located on the user's local media library. For example, the user may be required to pay for files located on an on-demand, streaming media service. Thus, the user may be required to pay a price for each media file before downloading or burning the media files. In another example, the user may be required to be subscriber to a service that allows the user to download and burn media files. In a further embodiment, the user can perform any of the above managing functions in any combination as the user desires using the media management program.

[0069] In a further embodiment, as depicted in **FIG. 3**, other programs or applications and multiple users **370** can access the playlist program **345** and the playlist storage **355** to search for playlists from non-media management applications. For example, other Yahoo! applications such as Yahoo! Search could access the playlists and enable users to create, edit, and share playlists through the Search user interface.

[0070] In some embodiments, the playlist entries within the playlist storage **355** and/or playlist index **375** are updated with all of a user's newly created or edited playlists. In one embodiment, it is important to keep the playlist storage and/or playlist index currently updated with all the newly created playlists and any edits made to the existing playlists via a user's local processor. In one embodiment, synchronizing all playlist edits and newly created playlists among each user's local processor and the playlist storage and/or playlist index requires both that each playlist be associated with an easy identifiable id and a technique for noting whether each particular playlist has been synchronized. For example, in one embodiment, once a playlist is created, it is associated with an id. In one embodiment, the id is generated by an online media service. Thus, the playlist is recognizable throughout the service via its associated id. In further embodiments, other online communities or media services may have access to these playlist ids in order to catalogue, identify, and/or otherwise interface with the playlists by their associated id.

[0071] In one embodiment, the list of playlists and its associated ids are maintained in a database **380** stored on a user's local processor **300** as depicted in **FIG. 3**. In one embodiment, the local database communicates with all of the components of the local processor **300** including the media engine **305**. Further, in one embodiment, this local database **380** also maintains a status list or status table comprising each playlist, its associated id, and a flag, marker, or other type of identifier indicating whether that playlist has been fully updated with respect to the playlist storage **355** and/or playlist index **375**. In one embodiment, the status table comprises only two states, updated and not-updated. In one embodiment, a playlist is considered not-updated if any type of edit has been made to a particular playlist, or a new playlist has been created and the playlist has not been synchronized with the playlist storage **355** and/or playlist index **375**. Thus, the local media engine **305** can reference the database's status table to determine whether a particular playlist needs to be synchronized with playlist storage **355** and/or playlist index **375**.

[0072] An exemplary synchronization process is depicted in **FIG. 11**. An exemplary system architecture or configuration supporting the synchronization process of **FIG. 11** is depicted in **FIG. 3**. Referring to **FIG. 11**, the synchronization process may be triggered **1105** based on a particular time interval such as every 15 minutes or every 30 minutes. In another embodiment, the synchronization process may be triggered **1105** by an event such as starting the playlist application. In yet another embodiment, the synchronization process may be triggered when the user changes any part of a playlist.

[0073] Once the process is triggered **1105**, with reference to both **FIG. 11** and **FIG. 3** the local media engine **305** will request the local database **380** to upload to the playlist storage **355** and/or playlist index **375** any playlists that are flagged as not updated. In another embodiment, a playlist is flagged as updated when a user creates a new playlist or when the user modifies an existing playlist. If the upload is successful **1115**, the local media engine **305** flags the each successfully uploaded playlists as updated. In one embodiment, if the upload is not successful **1120** or partially successful, each unsuccessfully uploaded playlist will remain flagged as not updated.

[0074] In some embodiments, multiple processors at different locations may be utilized by a single user and/or subscriber to an online media service. An exemplary system configuration comprising multiple processor ids depicted in **FIG. 13**. As depicted in **FIG. 13**, the system configuration has the same configuration and functionality as in **FIG. 3**, however with an additional processor **1300** and user **1330**. For example, a user may have a processor at home **300** and a processor **1300** at work and would like to access all updated playlists from each processor **300**, **1300**. Thus, it is desirable to further synchronize the edited playlists and any new playlists among each of the processors **300**, **1300**. In one embodiment, media server **360** and/or playlist storage **355** maintains a list of each user's processor wherein each processor is associated with an id. For example, with reference to **FIG. 13**, processor **300** has an associate id of "ABC" and processor **1300** has an associated id of "DEF." In a further embodiment, this list is maintained on the playlist program **345**, and/or playlist index **375**. In one embodiment, when a playlist is updated and uploaded to the media server **360** and/or playlist storage **355** as depicted in **FIG. 13**, the media server **360** and/or playlist storage **355** will note which processor, via processor id, sent the uploaded playlists. Thus, the media server **360** and/or playlist storage **355** will know which processor ids have the uploaded and updated playlists and which processor ids do not have the uploaded and updated playlists. An exemplary synchronization process is depicted in **FIG. 12**. An exemplary system architecture or configuration supporting such a synchronization process is depicted in **FIG. 13**. It should be noted that this process applies to any number of processors and the descriptions are merely exemplary with respect to processor **1300**. Further the actions performed by the processor **1300** as **FIG. 12** and FIG. **13** can also be performed by the media engine **1305**. Referring both to **FIG. 12** and **FIG. 13**, first, the synchronization process is triggered **1205**. In one embodiment, the process may be trigged by a pre-determined, configurable time interval or by an event such as a user logging onto a processor. Once the process is triggered **1205**, the local processor **1300** requests **1210** from the playlist storage **355** to download any newly edited or newly created playlists to the local processor **1300**. Further, in one embodiment, this request **1210** may or may not be facilitated by the media server **360** and/or playlist program **345** as depicted in **FIG. 13**.

[**0075**] Upon the request **1210**, the playlist storage **355** and/or playlist index **375** references the requesting processor's **1300** id to determine whether it already has the updated playlists. If the processor **1300** already has the updated playlists, then the playlist storage **355** and/or playlist index **375** will not need to download **1220** any information to the requesting processor **1300**. If the processor **1300** does not have all the updated playlists, the playlist storage **355**, and/or playlist index **375** sends **1225** a list of the playlist ids to be downloaded to the requesting processor **1300**. In addition, when the list of playlist ids is downloaded to the requesting processor **1225**, the local processor **1300** flags each downloaded playlist as not-updated. Thus, when appropriate, the local processor will synchronize the appropriate playlists as depicted in **FIG. 11** and when successfully downloaded, will flag each playlist as updated.

[**0076**] Playlists are not limited to music files and can include other audio or video files. For example, this system could be used to build playlists of music videos or to develop playable lists of music, audio news, commentary, rrs feeds, podcasts, still images, and video content. In a future application, users could be given tools to syndicate a playlist as an audio or video podcast.

[**0077**] Those skilled in the art will recognize that the method and system of the present disclosure within the application may be implemented in many manners and as such is not to be limited by the foregoing exemplary embodiments and examples. In other words, functional elements being performed by a single or multiple components, in various combinations of hardware and software, and individual functions can be distributed among software applications at either the client or server level. In this regard, any number of the features of the different embodiments described herein may be combined into one single embodiment and alternate embodiments having fewer than or more than all of the features herein described are possible. Functionality may also be, in whole or in part, distributed among multiple components, in manners now known or to become known. Thus, myriad software/hardware/firmware combinations are possible in achieving the functions, features, interfaces, and preferences described herein. Moreover, the scope of the present disclosure covers conventionally known and features of those variations and modifications through the system component described herein as would be understood by those skilled in the art.

We claim:

1. A method of identifying a published playlist comprising:

receiving, from a searching entity, search criteria representing a desire to identify a playlist;

accessing a playlist storage, said playlist storage comprising user published playlists, some or all of said published playlists comprising associated user-defined playlist descriptor information, said user-defined playlist descriptor information comprising user developed text;

searching said playlist storage to identify a desired playlist result set comprising user published playlists with user-defined playlist descriptor information relevant to said search criteria; and

communicating said desired playlist result set to said searching entity.

2. The method of claim 1 wherein said searching entity comprises a user located on a computing device.

3. The method of claim 1 wherein said searching entity comprises a software program.

4. The method of claim 1 wherein said search criteria comprises a mood associated with said desired playlist.

5. The method of claim 1 wherein said search criteria comprises a situation associated with said desired playlist.

6. The method of claim 1 wherein said search criteria comprises an event associated with said desired playlist.

7. The method of claim 11 wherein said playlist storage is located on a server.

8. The method of claim 1 wherein said playlist storage is located on the internet.

9. The method of claim 1 wherein said playlist storage is located as a part of an online media service.

10. The method of claim 1 wherein said desired playlist result set comprises a playlist identifier.

11. The method of claim 10 wherein said identifier is a uniform resource locator.

12. The method of claim 10 wherein said identifier is a pointer.

13. The method of claim 10 wherein said identifier is a universal resource indicator.

14. The method of claim 1 wherein said desired playlist result set comprises metadata associated with said desired playlist results set.

15. The method of claim 1 wherein said desired playlist results set comprises a permission level for accessing said desired playlist results.

16. The method of claim 15 wherein said permission level is associated with an online media service.

17. The method of claim 1 wherein said desired playlist results set comprises an id associated with said desired playlist.

18. The method of claim 1 further comprising facilitating the display of said desired playlist result set at a computing device associated with said searching entity.

19. A method of creating a searchable playlist comprising:

receiving at least one media file selection indicating the desire to include at least one media file in a playlist;

receiving user-defined playlist descriptor information associated with said playlist, said user-defined playlist descriptor information comprising user developed text;

receiving an input indicating the desire to designate said playlist as public, said public designation causing said user-defined playlist descriptor information and information associated with said playlist to be searchable by a searching entity; and

communicating to a playlist storage, said user-defined playlist descriptor information and said associated playlist information so that said user developed text can be searched by a searching entity.

20. The method of claim 19 wherein said playlist storage is located on a server.

21. The method of claim 19 wherein said playlist storage is located on the internet.

22. The method of claim 19 wherein said playlist storage is located as a part of an online media service.

23. The method of claim 19 wherein said published playlist information comprises information associated with at least one media file.

24. The method of claim 23 wherein said playlist information comprises the title associated with said media file.

25. The method of claim 23 wherein said playlist information comprises the artist associated with said media file.

26. The method of claim 23 wherein said playlist information comprises the genre associated with said media file.

27. The method of claim 23 wherein said playlist information comprises a rating associated with said media file.

28. The method of claim 19 wherein said searching entity comprises a user.

29. The method of claim 19 wherein said searching entity comprises a software program.

30. The method of claim 19 wherein said user developed text is searchable by search input criteria.

31. The method of claim 30 wherein said search input criteria comprises a mood associated with said playlist.

32. The method of claim 30 wherein said search input criteria comprises a situation associated with said desired playlist.

33. The method of claim 30 wherein said search criteria input comprises an event associated with said desired playlist.

34. The method of claim 19 wherein said user defined text comprises a mood associated with said playlist.

35. The method of claim 19 wherein said user developed text comprises a situation associated with said playlist.

36. The method of claim 19 wherein said user developed text comprises an event associated with said playlist.

37. The method of claim 19 wherein said user-defined playlist description information and said associated playlist information comprise a format comprising a set of searchable field entries.

38. The method of claim 37 wherein said set of searchable field entries comprise a portion of said user developed text.

39. The method of claim 37 wherein said set of searchable field entries comprise all of said user developed text.

40. The method of claim 37 wherein said set of searchable field entries comprise a portion of said associated playlist information.

41. The method of claim 19 further comprising synchronizing, upon receiving an edit to said playlist, said user-defined playlist descriptor information and said associated playlist information with said playlist storage.

42. The method of claim 41 wherein said edit comprises an input indicating the desire to designate said playlist as private.

43. The method of claim 19 wherein said storing said user-defined playlist descriptor information and said associated playlist information into a searchable format comprises serializing said user-defined playlist descriptor information and said associated playlist information.

44. One or more computer-readable media having computer-readable instructions thereon which, when executed by one or more computing devices, implements the method of claim 19.

45. One or more computer-readable media having computer-readable instructions thereon which, when executed by one or more computing devices, implements the method of claim 1.

46. The method of claim 1 further comprising synchronizing, upon receiving an edit to said playlist, said user-defined playlist descriptor information and said associated playlist information with said playlist storage.

* * * * *