

```

package dsa_craft;
import java.util.*;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;

public class Music_player {
    static Scanner sc=new Scanner(System.in);
    static Map<String, List<Node>> playlists=new
HashMap<>();
    static Stack<Node> stk=new Stack<>();
    static Map<String,Integer>count=new HashMap<>();
    static class Node{
        Node prev;
        Node next;
        String name;
        String art;
        String cat;

        public Node(String name, String art, String cat)
{
            this.name = name;
            this.art = art;
            this.cat = cat;
        }
    }
    static Node head;
    public static void add(String name, String art,
String cat)
    {
        Node newnode=new Node(name,art,cat);
        if(head==null)
        {
            head=newnode;
            head.next=head;
            head.prev=head;
        }
        else
        {
            Node last=head;
            while(last.next!=head)

```

```

        {
            last=last.next;
        }
        last.next=newnode;
        newnode.prev=last;
        head.prev=newnode;
        newnode.next=head;
    }
}

public static void display()
{
    if(head==null)System.out.println("no");
    else
    {
        Node last=head;
        System.out.println(last.name);
        last=last.next;
        while(last!=head)
        {
            System.out.println(last.name);
            last=last.next;
        }
    }
}

public static void play(Node song)
{
    System.out.println("<=(0) "+song.name+" Playing
(1)=>");
    recentlyplayed(song);
    if(count.containsKey(song.cat))
    {
        int a=count.get(song.cat);
        a=a+1;
        count.remove(song.cat);
        count.put(song.cat,a);
    }else count.put(song.cat,1);
    System.out.println("Do you want to add this
"+song.name+" song in playlist ? yes=1 OR no=0");
    int pl=sc.nextInt();
    if(pl==1)
    {
        System.out.println("Enter the name of the
playlist in which do you want add :");
        String plname=sc.next();
    }
}

```

```

        addPlaylist(song,plname);
        displayPlaylists(plname);
    }
    System.out.println("\n"+song.name+"
Playing.....");
    System.out.println("prev:-1    pause:0    next:1
Stop:2");
    int n=sc.nextInt();
    if(n==1)
    {
        play(song.next);
    }
    else if(n==-1)play(song.prev);
    else if(n==0)
    {

        System.out.println("\ndo you want to play
your any playlist ??");
        System.out.println("Menu ");
        System.out.println("1.do you want to play
your any playlist\n2.Do you want to Play Song of your
choice");
        int i=sc.nextInt();
        switch(i)
        {
            case 1:play_playlist();
                    break;

            case 2:System.out.println("enter the name of
song do you want to play");
                    String songname=sc.next();
                    Node node=search_song(songname);
                    play(node);
                    break;
        }
    }
    else
    {

    }
}
}
public static void play_playlist()
{

```

```

        if(playlists.size()==0)System.out.println("NO
playlist exists");
        else {
            System.out.println("PLAYLISTS :");
            for (Entry<String, List<Node>> mapElement :
playlists.entrySet())
            {
                String key = mapElement.getKey();

                System.out.println(key);
            }
            System.out.println("enter the name of playlist do
you want to play");
            String plname=sc.next();
            if (playlists.containsKey(plname)) {
                List<Node> list = playlists.get(plname);
                int p=1;
                int j=0;
                while(p==1)
                {
                    System.out.println(list.get(j).name+"
Playing.....");
                    System.out.println("prev:-1    stop:0    next:1
remove:2");
                    int ch=sc.nextInt();

                    switch(ch)
                    {
                        case 1:
                            j=j+1;
                            if(j==list.size())
                            {
                                j=0;
                            }
                            else if(list.size()==1)
                            {
                                do
                                {
                                    System.out.println(list.get(j).name+"
Playing.....");
                                    System.out.println("Do you want to stop
playing ??");
                                    ch=sc.nextInt();
                                }
                                while(ch==1);
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }

    break;
    case -1:
        j=j-1;
        if(j== -1)
        {
            //
System.out.println(list.get(list.size()-1).name+"
Playing.....");
            j=list.size()-1;
        }

    break;
    case 0:p=0;break;
    case
2:removeSongFromPlaylist(plname,list.get(j).name);
    break;
    }
}
}
}
}
}
public static void recentlyplayed song(Node song)
{
    if(stk.contains(song)) {}
    else stk.push(song);

}

public static void stkprint()
{if(stk.size()==0)System.out.println("NO song
played recently");
else { for(Node s:stk)
    {
        System.out.println(s.name);
    }
}
}

public static void createPlaylist(String
playlistName) {
    playlists.put(playlistName, new
ArrayList<>());
}

```

```

        System.out.println("Playlist '" +
playlistName + "' created successfully.");
    }

    public static void addPlaylist(Node song, String
plname) {
        if (playlists.containsKey(plname)) {
            List<Node> list = playlists.get(plname);
            int i=binarySearch(list,song.name);
            if(i== -1)
            {
                list.add(song);
                playlists.put(plname,list);
                System.out.println("'" + song.name + "'
added to playlist '" + plname + "'.");
            }
            else System.out.println("'" + song.name
+ "' exists into playlist '" + plname + "'.");

        } else {

                createPlaylist(plname);
                addPlaylist(song,plname);
                //System.out.println("Playlist '" +
plname + "' created");
            }
        }

    public static void displayPlaylists(String plname)
{
        System.out.println("Your playlists
"+plname+" contains songs :");
        List<Node> list=playlists.get(plname);
        for(Node s:list)
        {
            System.out.println(s.name);
        }
    }

    static int binarySearch(List<Node> productList,
String productName) {
        int low = 0;
        int high = productList.size() - 1;

        while (low <= high) {

```

```

        int mid = low + (high - low) / 2;
        String midProductName =
productList.get(mid).name;
        if(midProductName.equals(productName))
        {
            return mid; // Product found at index mid
        }
        else if
        (midProductName.compareTo(productName) < 0)
        {
            low = mid + 1; // Search in the right
half
        }
        else {
            high = mid - 1; // Search in the left
half
        }
    }
    return -1;
}
public static Node search_song(String songname)
{
    Node last=head;
    if(last.name.equals(songname))play(last);
    else
    {
        last=last.next;
        int flg=0;
        while(last!=head)
        {
            if(last.name.equals(songname))
            {
                return last;
            }
            last=last.next;
        }
        if(flg==0)System.out.println("song not
found");

    }
    return null;
}
public static void removeSongFromPlaylist(String
playlistName, String sName) {

```

```

        if (playlists.containsKey(playlistName)) {
            if (playlists.get(playlistName) != null &&
playlists.get(playlistName).contains(sName))
            {
                List<Node>
list=playlists.get(playlistName);
                list.remove(list.indexOf(sName));
                System.out.println("Song '" + sName + "'
removed from playlist '" + playlistName + "'.");
            }else{
                System.out.println("Song '" + sName + "'
not found in playlist '" + playlistName + "'.");
            }
        } else {
            System.out.println("Playlist '" +
playlistName + "' not found.");
        }
    }

    public static void main(String[] args) {

add("Shabashiya","shweta","sad");
add("Parinda","gaury","motivational");
add("Khwab","sakshi","sad");
add("Chor","nikita","sad");
add("Aigirinandini","shraddha","classical");

int choice=0;
do {

    System.out.println("----- MENU
-----");
    System.out.println("1.Play\n2.Search\n3.play your
playlist\n4.view history\n5.Stop");
    System.out.println("Enter your Choice");
    choice=sc.nextInt();
    switch(choice) {
        case 1:
            play(head);
            break;
        case 2:
            System.out.println("Enter the song do you want
to listen");
            String s=sc.next();
            Node node=search_song(s);

```



```

        if(node==null) System.out.println("the song do
not exists");
        else play(node);
        break;
    case 3:play_playlist();
    break;
    case 4:stkprint();
    break;
    case 5:System.out.println("Thank you listening");
    choice=0;
    break;
}
}while(choice!=0);
for(Entry<String, Integer> entry: count.entrySet()) {

    if(entry.getValue()
==Collections.max(count.values())) {
        System.out.println("User has interest in : " +
entry.getKey() +" Category song . it has count : "
+Collections.max(count.values()));
        break;
    }
}

}

}
}

```