**PLAYLIST MANAGER DESIGN POINTS**

1. PlayList will have Songs , so we will Need a class Song , which can have details like track Identifier ,file name , track name etc. etc.
2. We will need a data structure to store songs in play list , General operations in a playlist are mostly (% estimates based on my usage ..so these are my assumptions)

* Add Song To End Of Play List (40% of times)
* Delete Songs (20%)
* Repeated songs in a play list (20-25 %)
* Play a particular song (depends on usage for me mostly 10% of times , you will usually let songs play one after another in play list and **at most you will do next if you don’t like.**

This makes two kind of data structures a good choice for playlist

1. Vector
2. Linked List

For a vector , we have O(1) access to a song , but we will need to resize and reset a lot of indices , Vectors have good support for operations like insert at and removeAt in form of Vector from C++ STL and List in C# , so easy to write neat and maintainable code with Vectors

For a linked list , we have insert at end and start in O(1) , access to a particular song is O(n), insert and delete operations are O(n) , shuffle will need extra space of O(n) to shuffle

Both of these are good candidates for data structures, in addition to this we will maintain a cache in form of Dictionary (C#) and unordered\_map C++ (STL) , this will contain mapping from track identifier to a song.

**I have implemented in this solution C++ Console application, C# class library , C# console application and a unit Test project to test Library. In C# I have implemented playlists using both the data structures Linked list and vector and in c++ I have used Vectors.**

**InterFace : IPlayList**

**Contains signature for create , insert, delete , play , print and shuffle methods.**

**Contains two properties collection of songs and currentPlayingTrack (helpful in testing code and writing UTS also)**

**Two classes in C#**

**PlaylistVector – uses C# list**

**PlayList – uses C# LinkedList**

**Class in C++**

**Playlist –uses vector from STL**

**Regarding Console interface, in case of invalid command or exception error is thrown.**