10/31/2019

Mark Xie

South Metro TAFE

Project Planning Document

Programming

Contents

[1. What data structures are you using? 2](#_Toc26519865)

[What is Dynamic Array 2](#_Toc26519866)

[List<T> 2](#_Toc26519867)

[2. Where are you using hashing techniques? 2](#_Toc26519868)

[What sorting algorithm are you using how this is different from selection and bubble sort? 2](#_Toc26519869)

[Bubble sort 3](#_Toc26519870)

[Selection sort 3](#_Toc26519871)

[3. What search technique are you using? 3](#_Toc26519872)

[Linear Search 3](#_Toc26519873)

[One reason I think linear search is a good option is because many people like how they adding music and they want to keep it unsorted. This disadvantage of using binary search is that when you using binary search you have to sort it first and binary search needs a sorted array. 3](#_Toc26519874)

[4. What third party libraries are you using? 4](#_Toc26519875)

[5. Where can I find the documentation for this? 4](#_Toc26519876)

[6. A mock-up of the GUI. 5](#_Toc26519877)

[7. What source control are you using? 5](#_Toc26519878)

[8. What are your coding standards you are enforcing? 5](#_Toc26519879)

[9. What tests are you going to run? 6](#_Toc26519880)

[Test Case 6](#_Toc26519881)

# What data structures are you using?

## What is Dynamic Array

The dynamic array provides dynamic memory allocation, adding, searching, and sorting elements in the array. Dynamic array overcomes the disadvantage of the static array. In a static array, the size of the array is fixed but in a dynamic array, the size of the array is defined at run-time. List<T> is the dynamic arrays in C#.

## List<T>

List<T> class represents the list of objects which can be accessed by index. It comes under the System.Collection.Generic namespace. List class can be used to create a collection of different types like integers, strings etc. List<T> class also provides the methods to search, sort, and manipulate lists.

I have been using doubly linked list as my data structure during AT2. linked list is a linear data structure which is pretty good for a project like Media player because it helps with built-in method like Add Last, Add First, Next and previous. But it is kind difficult to use and does not have a good index based methods. Example, I have tried to get the index when selected ListBox is clicked, for linked list you require to use ElementAt and you need to write lot of code to get the index for linked list that add to the media player.

For the final projects I want to use List<string> as my data structure. Major reason is because it offers lot more useful method like linked list and has some great index based methods you can use.

# Where are you using hashing techniques?

I would be using hashing techniques to hash password to protect it against attackers who gain read-only access to the database.

One way of doing this is when user finished his register, the method hashes the password and add a random salt value into it, then adding user information with hashed value into database. When the user try to log in to the musicplayer application it will read username from the user and check into database, if database find the username it been stored in the database it will then read hash the password with the salt value that is stored in the database back to the application and has a method to validate passwords matches. If it does it will display the main music player application. Otherwise a message box will display with the information on it.

# What sorting algorithm are you using how this is different from selection and bubble sort?

I will use bubble sort as my sorting algorithm. Because in most case users are not likely stores 10,000+ mp3 file to their playlist, which means simple sorting algorithm like bubble sort is a good option in this scenario.

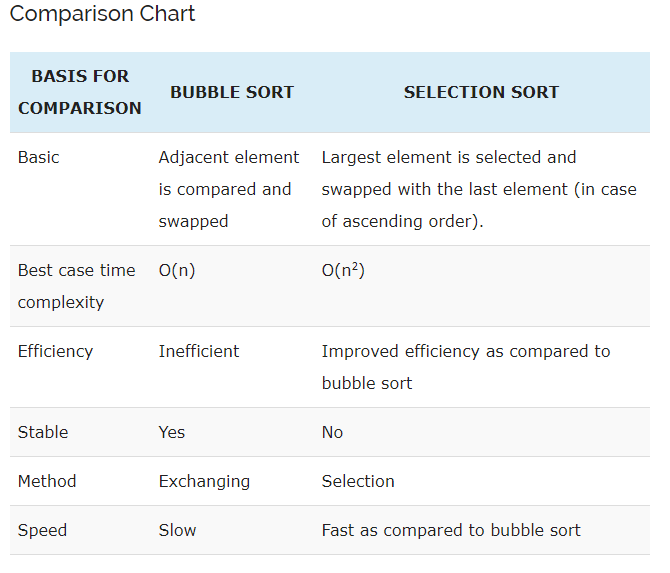
### Bubble sort

The bubble is one of the simplest sorting algorithms to understand and implement.it is O (n2) complexity means that its efficiency decreases dramatically on a //lists of more than a small number of elements.

### Selection sort

The selection sort is an algorithm that select the smallest elements [0] from the unsorted list. Each time it sorts the elements it swaps the elements and it keep the sorted list at beginning of the unsorted list.

Difference between the two is that bubble sort is stable algorithm (most stable and fast algorithms use additional memory) while selection sort is an unstable algorithm.



# What search technique are you using?

## Linear Search

A linear search scans one item at a time, without jumping to any item.

The worst case complexity is O (n), sometimes known an O (n) search. Similar to bubble sort, the time taken to search elements keep increasing as the number of elements are increased.

## 

## One reason I think linear search is a good option for this Project is because many people does not want to sort their music playlist and they want keep playlist unsorted. This is a huge disadvantage of using binary search because when you using binary search you have to sort it first and binary search needs a sorted array.

For example: when you trying to find number 3 in an unsorted list using binary search, the method starts at middle of an array . If the middle value is 11, the number you trying to find is at the last index. The binary search method will only looking for value at left side of the array as value 3 supposed to be smaller than value 11. Which is not going to find value 3 because it is at right-side of the array.

# What third party libraries are you using?

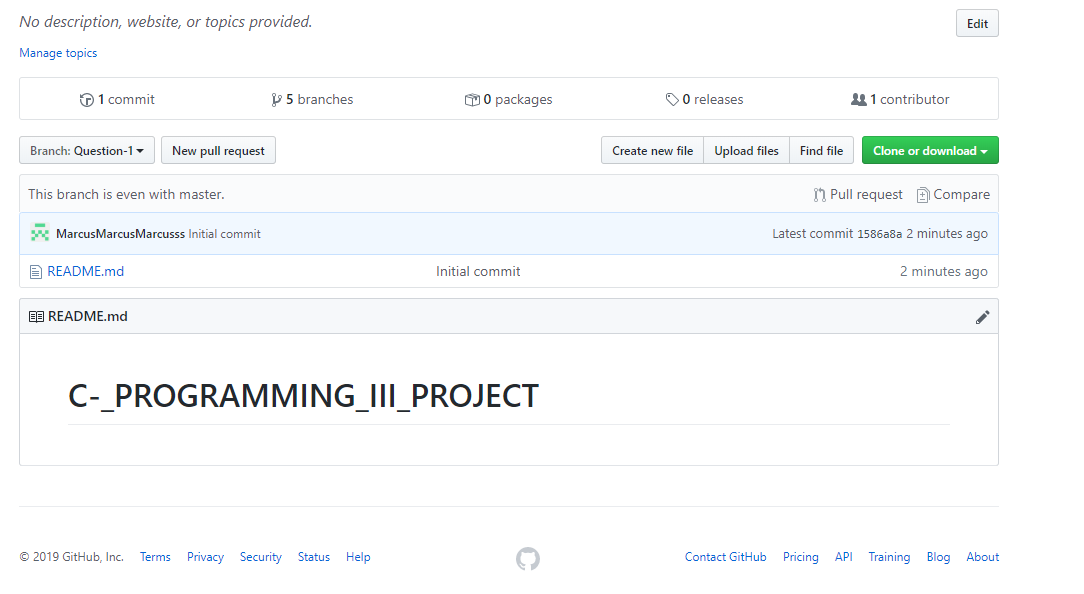
**MySQL**

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL).

MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands.

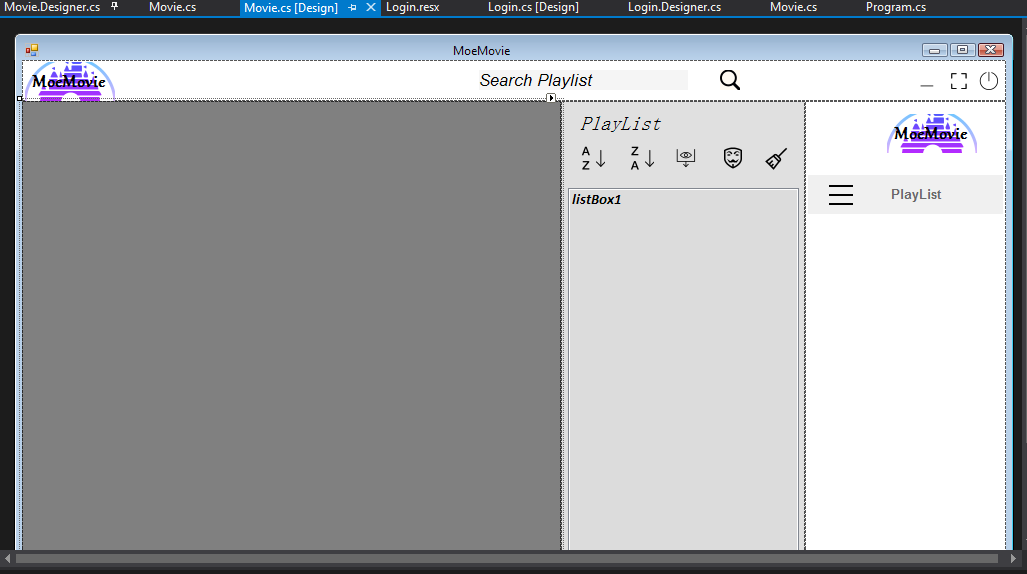
## Where can I find the documentation for this?

Nearly every open-source project uses GitHub to manage their project. Using GitHub is free if your project is open source and includes a wiki and issue tracker that makes it easy to include more in-depth documentation and get feedback about your project.



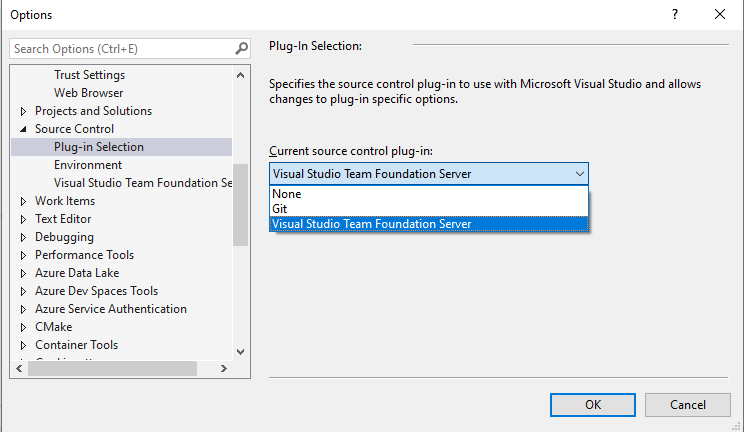
GitHub is an open source management tool and you can find it in here: <https://github.com/MarcusMarcusMarcusss/C-_PROGRAMMING_III_PROJECT>

## A mock-up of the GUI.



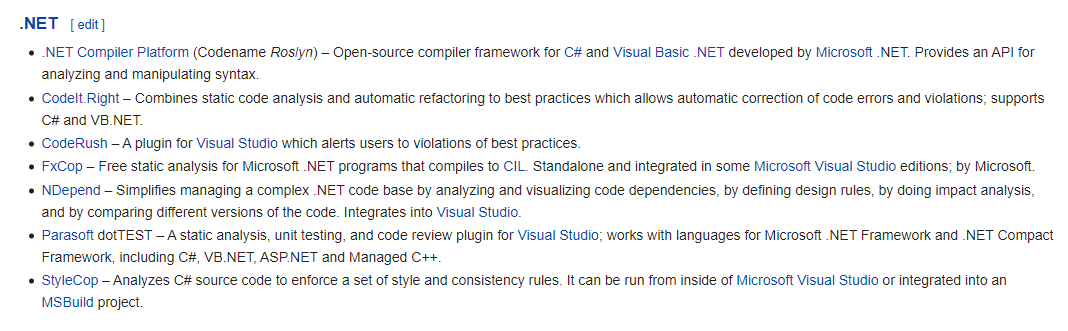
## What source control are you using?

As I am using visual studio 2019. It is already the latest version and has a Source Control Explorer window: From the Team Explorer home page (Keyboard: Ctrl + 0, H), choose Source Control Explorer. Or from the menu bar. You can choose Git or GitHub, both can be accessed in the tools-> options and then navigate to source control option.



## What are your coding standards you are enforcing?

 Windows visual studio Winforms- Microsoft's .NET Framework 4.7.2



## What tests are you going to run?

### Test Case

A Test Case is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, and post condition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

Test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Example of test case:

1. Test Case 1: Check results on entering valid User Id & Password
2. Test Case 2: Check results on entering Invalid User ID & Password
3. Test Case 3: Check response when a User ID is Empty & Register Button is pressed, and many more