****

**Data Structures**

**CS218**

***Project report***

**Group Members:**

Gulsher Khan 19I-0445

Abdurrafay 19I-0610

Muhammad Ameen 19I-0690

**Section:** C

**Distributed Hash Table**

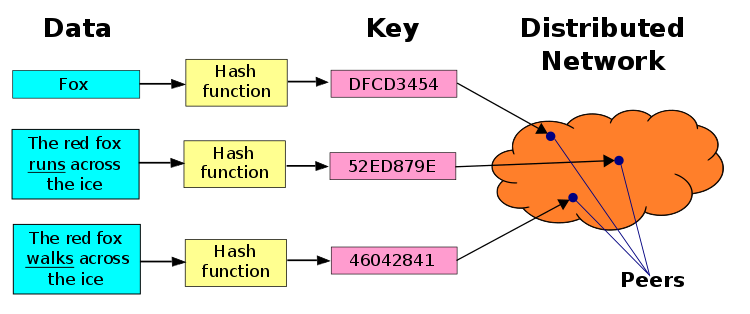
Introduction:

In this fast-growing field of computer science and the development and even more rapidly growing sectors of new scenario-based systems, new technology, and the future is really hard to predict. We are working with this Scenario based system as final project of Data structures course with 4 credit hours including lab, and as part of our degree this project helped us in better understanding of different components of Data structures.

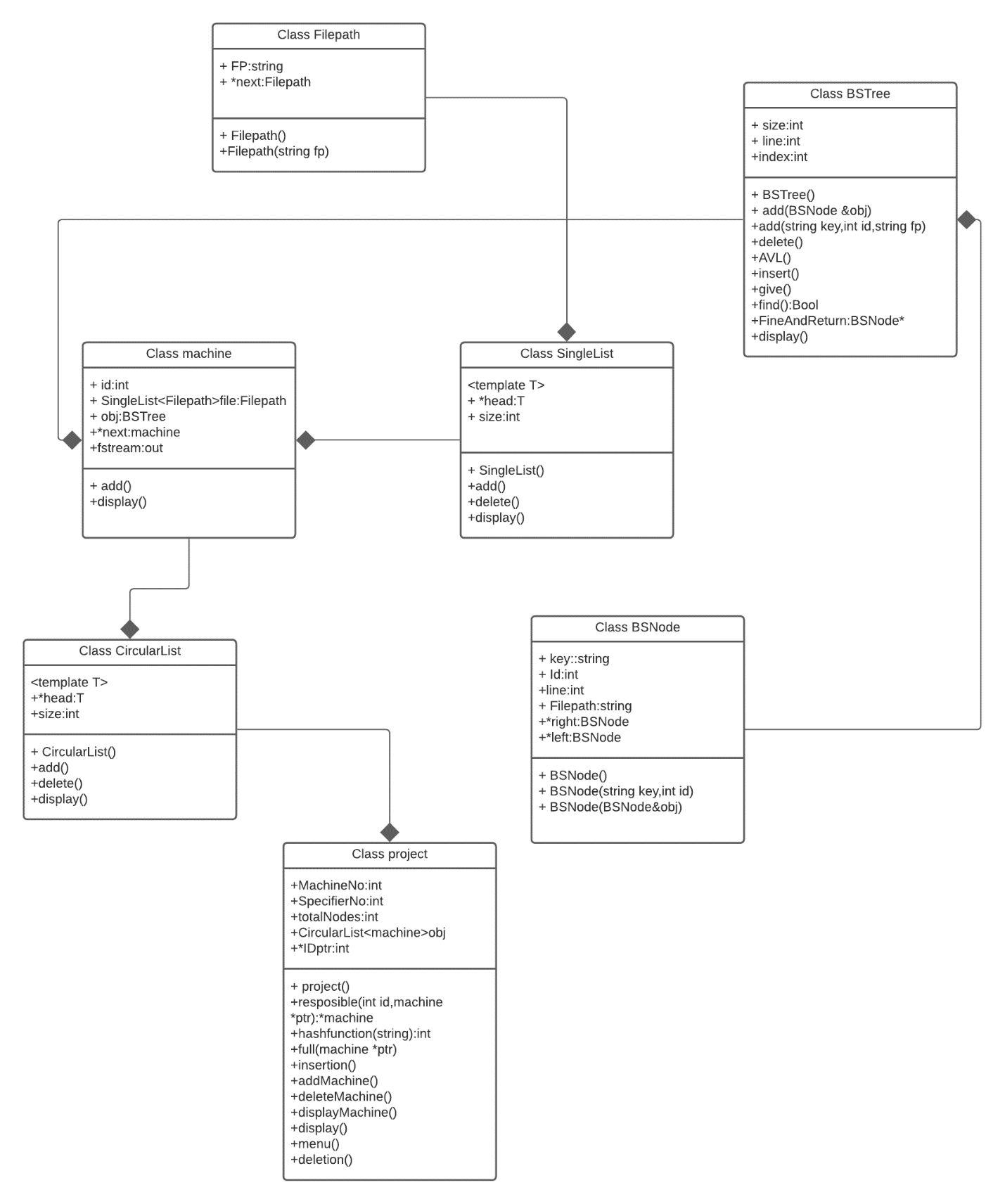
Project Description:

We were given a scenario where data is located on different machines geo distributed across internet. We were required to implement the searching and retrieval of data in such a scenario considering DHT (Distributed Hash Table).

Here is the DHT figure (source Wikipedia)



Class diagram:



**A Brief overview of Working:**

Main class is the **project class**. The **project class** is using all of the other classes to create the functionality of our Project. The **project class** contains all the important functions that are required in the Project. First of all, when an object of **project class** is made it will call on the **constructor** of **project class**. This **constructor** shall take values from the user such as the bit size of the system along with the total number of machines that the user wishes to have in the system. This **constructor** will also give the user the option to give ID to the machine manually, but if the user does not wish so, then the **constructor** will give the machines ID randomly. Then the **constructor** shall make a **circular linked list** of the machines. The **constructor** will also give the machines their first file (in which they shall store data). The rest of the files, if need arises to be created, will be made later in the code during the insertion process.

After this has been done, the **menu function** of the **project class** will be called. This function will give the user a menu to decide from what to do.

It shall have the following options:

1. Display the ID of all the machines.
2. Display the AVL of all machines.
3. Insert data in the system.
4. Delete data from the system.
5. Add machine in the system.
6. Delete machine from the system.
7. Exit

**Display ID of machines:**

This function shall display all the ID of all the machines.

**Display AVL of all machines:**

This function shall display the AVL of each and every machine.

**Insert data in system:**

This function will give the user two choices:

1. To enter the data in the form of key and value manually.
2. Or to specify a .txt file in which all the data in placed line by line in key value format. The program shall then read that whole file and then insert it into the system.

Now that we have the data, the key shall pass through the **Hash Function** and told an ID. A **responsible function** shall then be called which will tell which machine is responsible for this entry. From then on this data shall go to the AVL of that machine as well as the file of the machine. Before each insertion is made, a **full function** shall be called to check whether if the file of that machine is full or not (if it has a 100 entries then it is full). If it is full then the machine will be given another file. All the machines shall have a **singli-linked list** of their respective files.

**Delete data from system:**

This function shall take a key and check which machine is responsible for it through the **responsible function**. After it has found the responsible machine it shall check if the machine has the key in it’s AVL or not through the **find function**. If the machine does not have the entry, then it shall declare such. But if it does then it will delete that entry both from the file and the AVL. The AVL will always balance itself.

**Add machine in system:**

This will add a machine in the circular linked list of machine. The user can manually give the machine an ID or the system can give it a ID randomly.

**Delete Machine:**

This function will delete the machine from the circular linked list of machines and also will delete the AVL of the machine.

**Exit:**

This will exit the program