

|  |  |
| --- | --- |
| NU-Fast-University-Official-Logo.png | National University of Computer and Emerging Sciences  (Islamabad Campus)  Department of Computer Science |

**DATA STRUCTURES**

**Project Report**

**Project Title: Distributed Hash Table (Simulation of distributed system of machines**

**Group:**

**Abdul Hamid (19i-0565)**

**Rohan Saeed (19i-0411)**

**Section: CS-E**

**SUBMITTED TO: Data Structures Goggle Classroom**

**Date of Submission:-27th** December, 2020

The objective is to simulate a distributed system of machines where each machine has a unique id. The distributed system is circular linked list of machines. Every machine has a file attached to it where its data is stored. Data is given in the form of key-value pair. The key of the data is hashed using SHA-1 and a numeric id is obtained based on which the machine responsible for it is identified and the data is stored accordingly.

Each machine has an AVL tree maintained, each node of which stores a key, a file path, and line number where the value of the key is stored. Each machine also have a routing table which is a doubly linked list and it stores the successors of the machine according to formula p+2^i-1. The routing table helps to reduce time complexity of searching a machine from O(N) to O(log N).

The machines are allocated ID’s based on ip address/name as well as manually. The number of machines is specified at the beginning however any time latter a machine can be added and removed without disturbing the distributed system.

The ID of machines and the numeric hashed value of each key can be in range of 0 to 2^(identifier space)-1. Identifier space is specified in the beginning in term of number of bits.

Hashing is done in the following way:

f

Store in unsigned integer type

Convert to decimal

Take least significant 32 bits

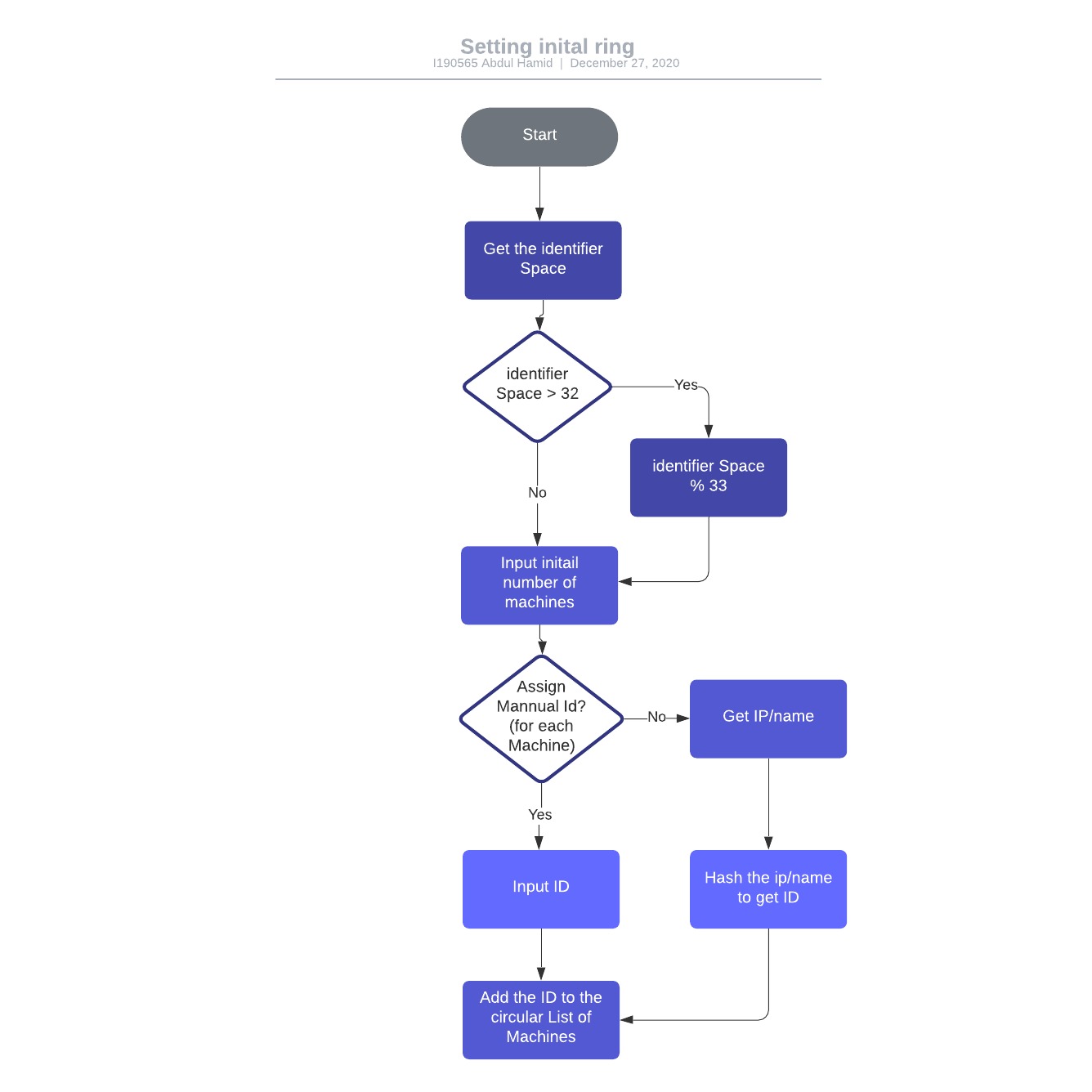
160 bit binary string

40 digit hexadecimal string

SHA-1

Input String

Initial setup of the machines ring is done in the following way:

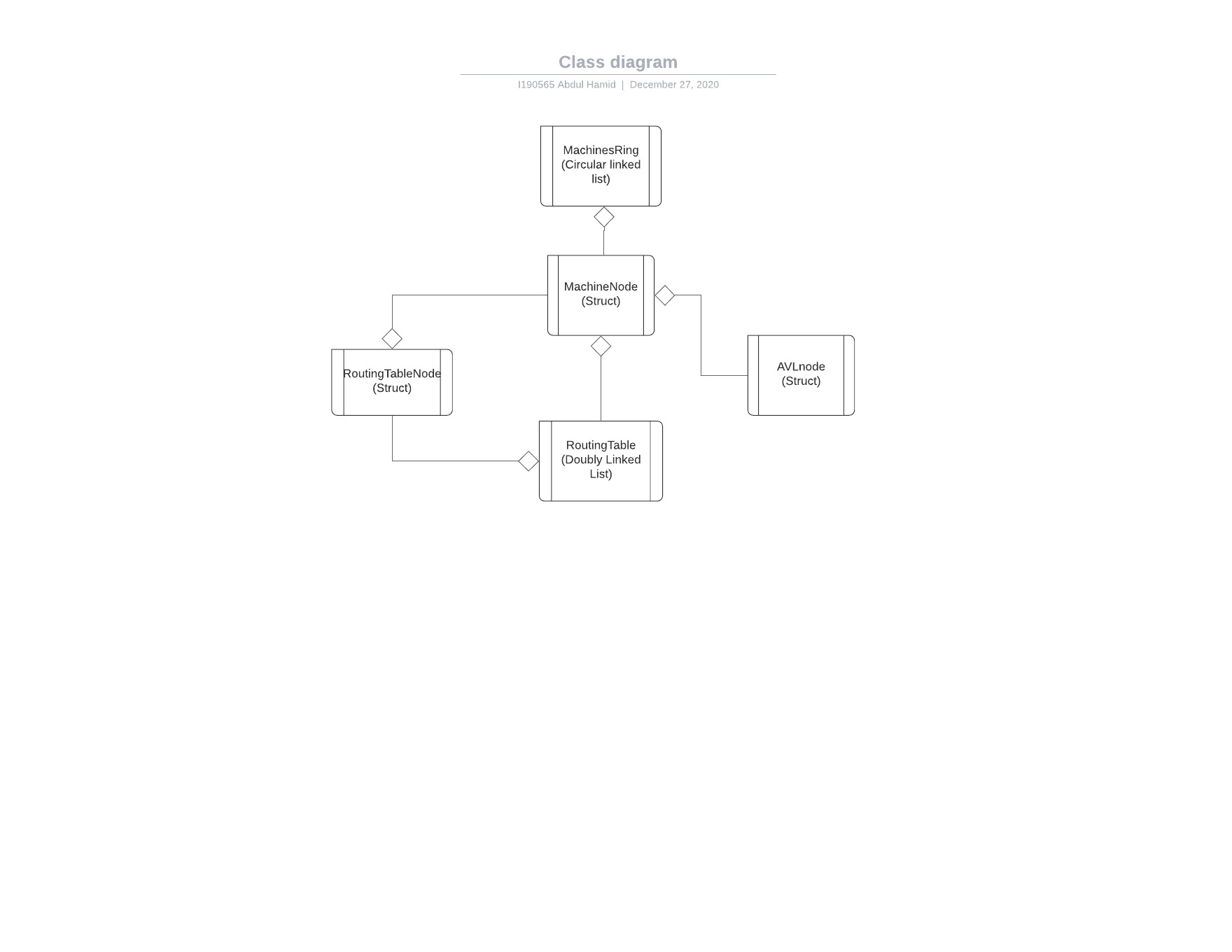


After machines are added data can be added from any machine. The key is hashed then the responsible machine is identified, in the avl tree the key the added the line line number is added and the value is stored in file in the specified line number. Data can also be removed from any machine, the key is hashed and the corresponding machine is identified and the avl node of the key is found, then the data from the file is removed and the node is deleted.

The avl tree of any machine as well as the routing table of any machine can also be printed using the functions of the respective classes.

When a new machine is added or a machine is removed from the chain, the chain is re adjusted accordingly. When new machine is added, all the keys in its successor for which this new machine is responsible now are added to this. When a machine is removed its data is shifted to its successor.

The relation between classes is mostly aggregations as we have pointers of different structures and classes inside the others.



Note: All the following operations can be performed from any machine node.

Data is added in the following way:

Add the key to avl tree and value to file

jkhlg

Find the machine responsible

Input key and value

Hash the key

Data is removed in the following way:

Find corresponding AVL node

Find corresponding machine

Hash the key

Key is input

Remove the value from file using line number

Delete this node

New Machine is added in the following way:

The respective nodes from the successor are deleted

Data form the successor whose key now belong to new machine are taken into it

ID is added to machines ring

ID is taken manually or hashing IP

Routing tables are recalculated

A machine is removed in the following way:

Find the machine

Shift its data to successor

And delete its file

Delete the machine from the ring

ID is input

Recalculate Routing tables