Names: Mujtaba Ali, Asfandyar Sabri, Muhammad Uzair
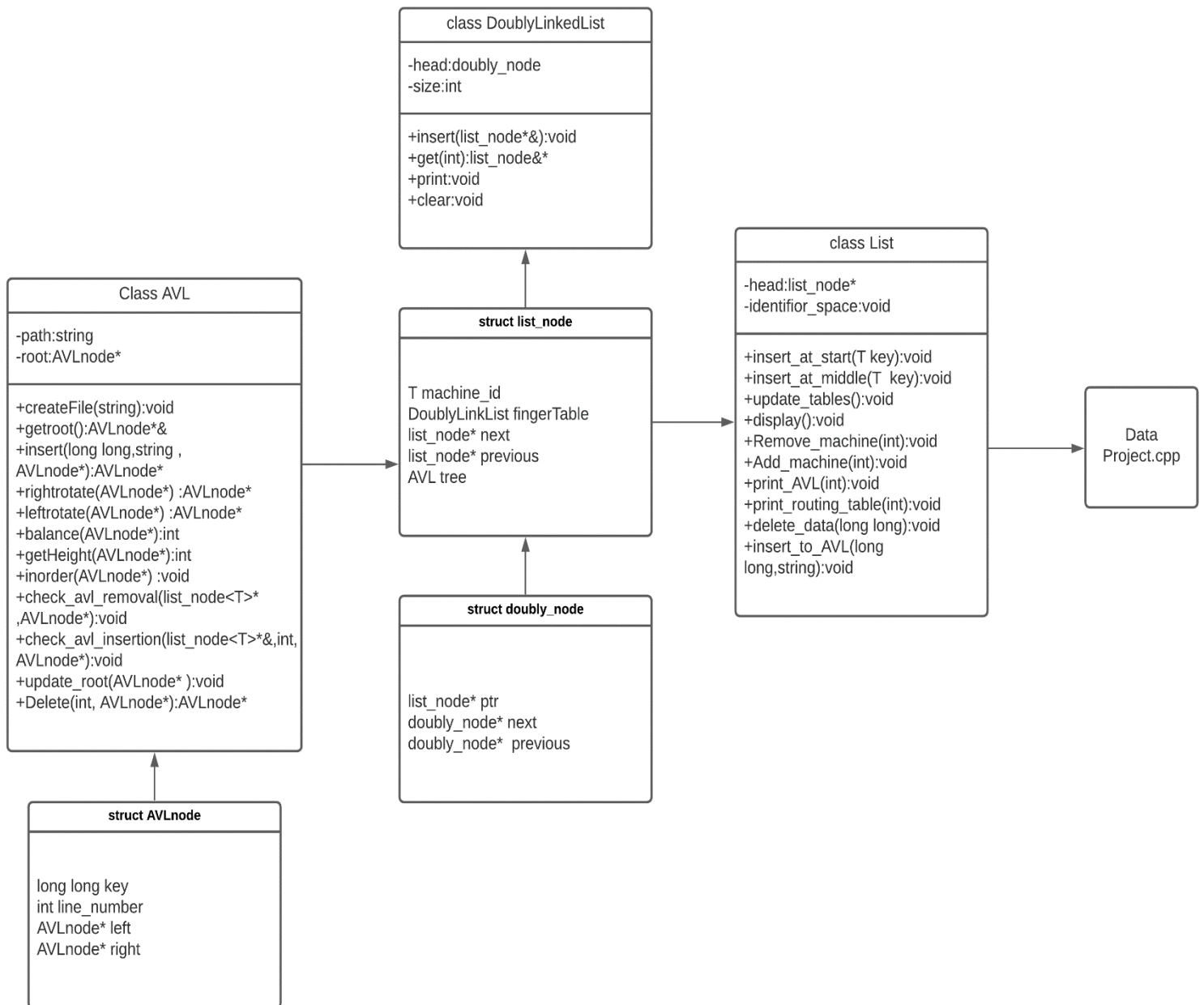
Section: CS-E

Roll#s: 19I-0577,19I-0410,19I-2187

DS PROJECT REPORT AND DOCUMENTATION

CLASS DIAGRAM

class DoublyLinkedList

-head:doubly_node
-size:int

+insert(list_node*&):void
+get(int):list_node&*
+print:void
+clear:void

Class AVL

-path:string
-root:AVLnode*

+createFile(string):void
+getroot():AVLnode*&
+insert(long long,string ,
AVLnode*):AVLnode*
+rightrotate(AVLnode*) :AVLnode*
+leftrotate(AVLnode*) :AVLnode*
+balance(AVLnode*):int
+getHeight(AVLnode*):int
+inorder(AVLnode*) :void
+check_avl_removal(list_node<T>*
,AVLnode*):void
+check_avl_insertion(list_node<T>*&,int,
AVLnode*):void
+update_root(AVLnode*):void
+Delete(int, AVLnode*):AVLnode*

struct list_node

T machine_id
DoublyLinkList fingerTable
list_node* next
list_node* previous
AVL tree

class List

-head:list_node*
-identifior_space:void

+insert_at_start(T key):void
+insert_at_middle(T key):void
+update_tables():void
+display():void
+Remove_machine(int):void
+Add_machine(int):void
+print_AVL(int):void
+print_routing_table(int):void
+delete_data(long long):void
+insert_to_AVL(long
long,string):void

Data Project.cpp

struct doubly_node

list_node* ptr
doubly_node* next
doubly_node* previous

struct AVLnode

long long key
int line_number
AVLnode* left
AVLnode* right

Documentation and Report

At first the program asks user to specify identifier space and numbers of machine.

For those machines, the program has a List class which is a doubly circular list, each node of it represents a virtual machine.

Each node (Virtual Machine) has one **unique machine id**, an **object to fingerTable** (Ftp), an **object to an AVL tree**).

The machine id can be assigned manually or automatically. The program asks user for the choice, in manual the user can choose machine ids while in automatic the ids are assigned randomly then machines are created. These machines are connected to each other through a doubly circular link list.

The fingerTable is a doubly linked list. Fingertables are creating using the successor formula, one for each machine which gives a path to other machines depending which machine is responsible to store the nodes of identifier space generated from successor.

Tree is an AVL tree. Each machine has one AVL tree. Each node of it can store a key and a line number.

The user is given an option to store data in these machines, the entered data is first hashed through a hash function which generates a key. Then we mod the generated key with $2^{identifier\ space}$. The generated value after taking mod decides which machine is responsible to store that data. The key is inserted in AVL tree of that machine and the data is stored in a txt file, associated with that specific machine, the tree also stores the line number where that data is stored.

User has given option to keep on adding data or do other functionalities which are following

Remove specific data from key.
Print the routing table of specified Machine id.
Print the AVL tree associated with the Machine id.
Add a new Machine.
Remove an existing Machine.

When the above functions are performed nothing is disturbed, everything remains in order i.e.

When adding data, the data after taking mod of its key goes to the machine which is responsible. So, every machine only stores data which they are responsible for.

Removing specific data makes that line blank from file.

Print AVL tree function prints the tree in Inorder-Traversal, it prints the data, key and filePath.

Adding a new machine updates routing table like they are supposed to be and the data which now this new machine is responsible for will get into it and will leave from its previous file.

When an existing machine is removed the data from it goes to the files of machine which are responsible for those data and the file of this machine will be deleted.

The FingerTables are generating.
The data is inserting.
The data is removing.
New Machines are being added.
Existing Machines are being removed.

Everything is being done through FingerTables and the Paths taken for looking up nodes are defined.