## PROBLEM DEFINITION

The problem is to implement a print queue simulator using a linked list. The program should allow the user to add new print jobs, print a document, cancel a print job, and display all of the print jobs that are currently waiting to be printed. The simulation must contain these abilities via a console menu that allows the user to control the program. The data for the program should be stored in a linked list class with a structure that contains all of the print job information.

## ANALYSIS

To implement this print manager simulation I used a single class defined as the LindedList and a single structure defined as MyNode. The MyNode structure is used within the LinkedList class as a single node of the LinkedList. Using the class has allowed me to hide the logic of how the print queue works and allow it to be used in any situation. This gives the main function of the program using this LinkedList much cleaner code. This also allows the LinkedList class to be reused in other programs that may need a class with the same features as this one. The diagram of the class and node structure are below. The decision to use a linked list to implement this simulated queue allows the program to dynamically add and remove the nodes which will keep the memory use of the program to the bare minimum. This could be of great interest if such a program was used in a limited computing environment.

| **LinkedList** |
|---|
| -head: MyNode* |
| -tail: MyNode* |
| |
| +LinkedList() |
| +~LinkedList() |
| +InsertNode(string,int,int,int,int, int, string): void |
| +DeleteNode(string): bool |
| +SearchNode(string): MyNode |
| +PrintDocument(): MyNode |
| + ListAll(): int |

| **MyNode** |
|---|
| int sequence_number |
| string document_name |
| int month |
| int day |
| int year |
| int hour |
| int minute |
| string owner |
| string service_required |
| MyNode *next |

## DESIGN

```
main function start
    LinkedList class instance printJobs
    integers choice and total_nodes declared
    character garbage declared
    strings doc_name and user declared
    integers month, day, year, hour, minute declared

    do/while loop for menu until user chooses 5 option
        display the menu with options 1 – 5

        get user input for menu choice

        switch decision on user input
            user chose 1
                prompt for all input to create new print job
                gather input for all required InsertNode arguments
                call InsertNode member function on printJobs
                break

            user chose 2
                call PrintDocument member function printJobs
                break

            user chose 3
                prompt user for print job name to delete
                input print job name
                call DeleteNode with print job name on printJobs
                if DeleteNode returns true
                    print message to console the print job was removed
                else
                    print message to console the print job was already printed
                break

            user chose 4
                print out print list title
                print out print list header
                call ListAll on printJobs and store return value in total_nodes
                print message to console with the number of print jobs remaining
                break

            user chose 5
                print message to console thanking user for using programs
                break

            default
                print message to console for correct choices
                break
```

## IMPLEMENTATION

The program was completed using Visual Studio Code on a system running the KDE Neon Linux Distribution. The CPU is an AMD Ryzen 1700x. The source files were located on the grace.bluegrass.kctcs.edu server. I used two extensions called Remote FS and SSHExtension that allowed me to open the remote files in my local text editor. I used SSH to connect to the grace.bluegrass.kctcs.edu server in a separate terminal to compile and run the program. The compiler used was g++ version 5.4.0. The program was compiled using the following command: g++ -o main LinkedList.cpp project4.cpp.

I tested the program manually by entering several sample print jobs. After creating new print jobs in the system, I then printed, canceled, listed, added more, printed, canceled. I tried printing all jobs that I had added and then adding more after the list was empty. I also canceled jobs that were at the beginning, middle and end of the list. I tried as many combinations as I could between adding, canceling, printing, and listing. I did encounter several unusual situations. The first unusual behavior is that is the user enters anything other than a number for the menu selection the program enters an infinite loop of the menu being reprinted. Another unusual situation involves input as well. If the user doesn't follow the formatting suggested for input the program will hang waiting for further input. I have included a text file containing some of the testing information I entered into the program.