



Library Management

Brief Specification

13.11.2019

Vyanktesh Chandurkar
A00268808

Introduction

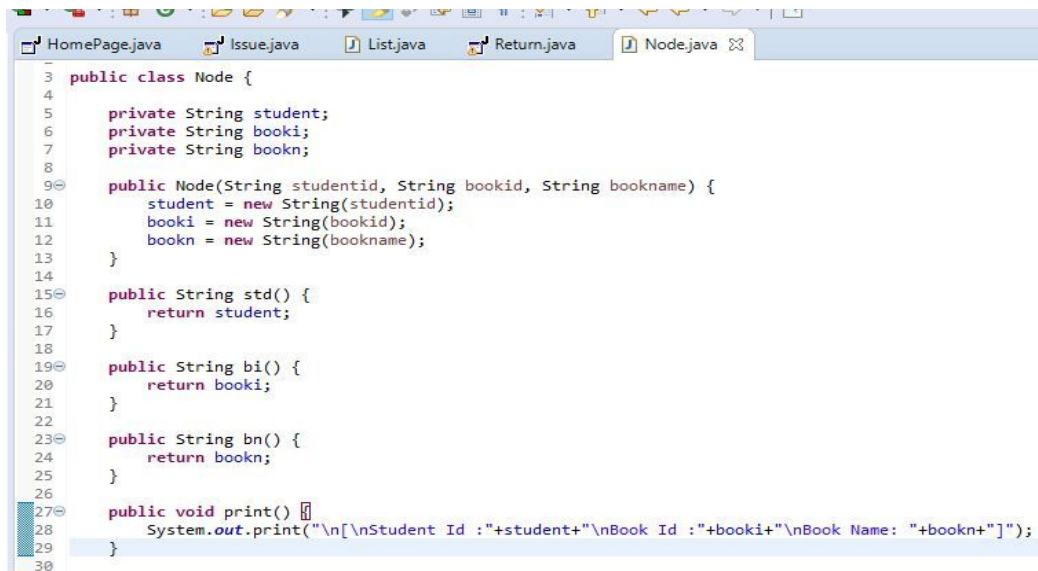
Library Management System will be using the concept of Array List. This application will make it easier to main library which will allow the user to issue and return books. It will save important information at the time of issuing the book like student ID, Book ID(Key) , Book name. It will be using insert and search function at the time of issuing and returning the book.

Classes:-

- Node - Contains structure of the node of the linked list.
- List - This contains all the push, pop and search functionalities of the application.
- Issue - This class is inside gui package and will be used when the user wants to issue the book.
- Return - This class is inside gui package and will be used when the user wants to return the book.
- Home - This class is inside gui package and will be used as a Home Screen

Code Snippets

- Node Class

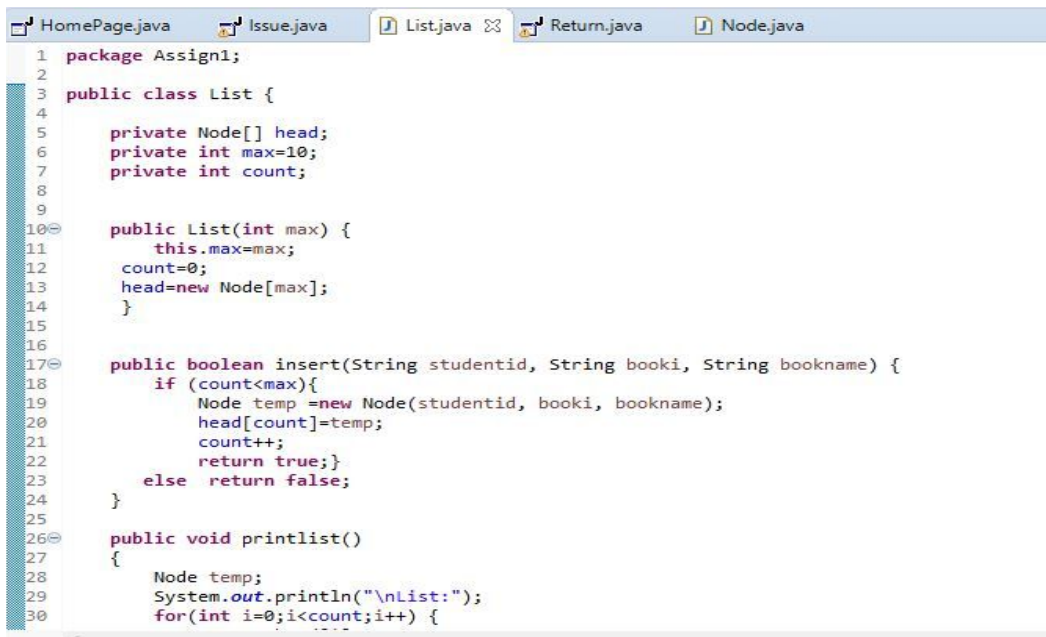


```

3 public class Node {
4
5     private String student;
6     private String booki;
7     private String bookn;
8
9     public Node(String studentid, String bookid, String bookname) {
10         student = new String(studentid);
11         booki = new String(bookid);
12         bookn = new String(bookname);
13     }
14
15     public String std() {
16         return student;
17     }
18
19     public String bi() {
20         return booki;
21     }
22
23     public String bn() {
24         return bookn;
25     }
26
27     public void print() {
28         System.out.print("\n[Student Id :"+student+"\nBook Id :"+booki+"\nBook Name: "+bookn+"]");
29     }
30

```

- List Class



```

1 package Assign1;
2
3 public class List {
4
5     private Node[] head;
6     private int max=10;
7     private int count;
8
9
10    public List(int max) {
11        this.max=max;
12        count=0;
13        head=new Node[max];
14    }
15
16
17    public boolean insert(String studentid, String booki, String bookname) {
18        if (count<max){
19            Node temp =new Node(studentid, booki, bookname);
20            head[count]=temp;
21            count++;
22            return true;}
23        else return false;
24    }
25
26    public void printlist()
27    {
28        Node temp;
29        System.out.println("\nList:");
30        for(int i=0;i<count;i++) {

```

```

16
17 public boolean insert(String studentid, String booki, String bookname) {
18     if (count<max){
19         Node temp =new Node(studentid, booki, bookname);
20         head[count]=temp;
21         count++;
22         return true;}
23     else return false;
24 }
25
26 public void printlist()
27 {
28     Node temp;
29     System.out.println("\nList:");
30     for(int i=0;i<count;i++) {
31         temp = head[i];
32         temp.print();}
33 }
34
35
36
37 public Node searchWithId(String booki){
38     Node res=null;
39     for(int i=0;i<count;i++){
40         Node temp=head[i];
41         if (temp.bi().equals(booki)){
42             res=temp;
43         }
44     }
45     return res;
46 }
47
48

```

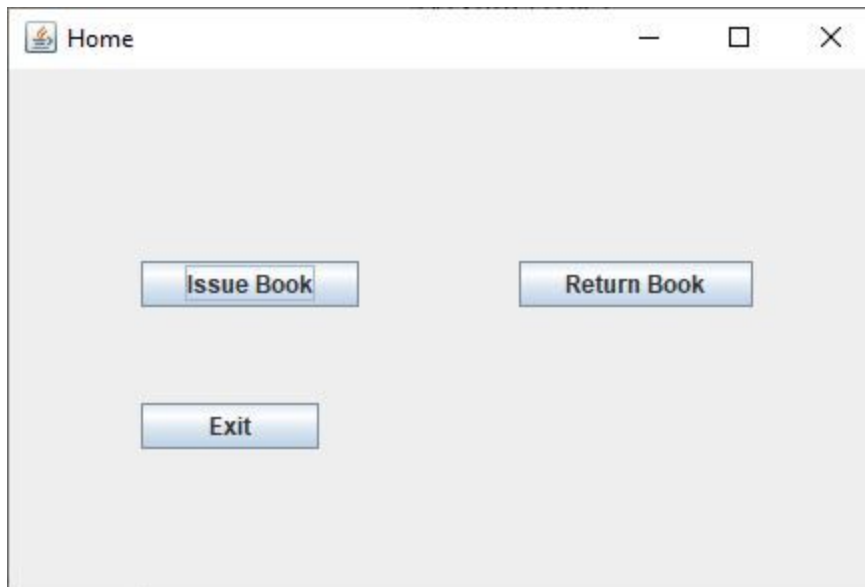
```

27
28 {
29     Node temp;
30     System.out.println("\nList:");
31     for(int i=0;i<count;i++) {
32         temp = head[i];
33         temp.print();}
34 }
35
36
37 public Node searchWithId(String booki){
38     Node res=null;
39     for(int i=0;i<count;i++){
40         Node temp=head[i];
41         if (temp.bi().equals(booki)){
42             res=temp;
43         }
44     }
45     return res;
46 }
47
48

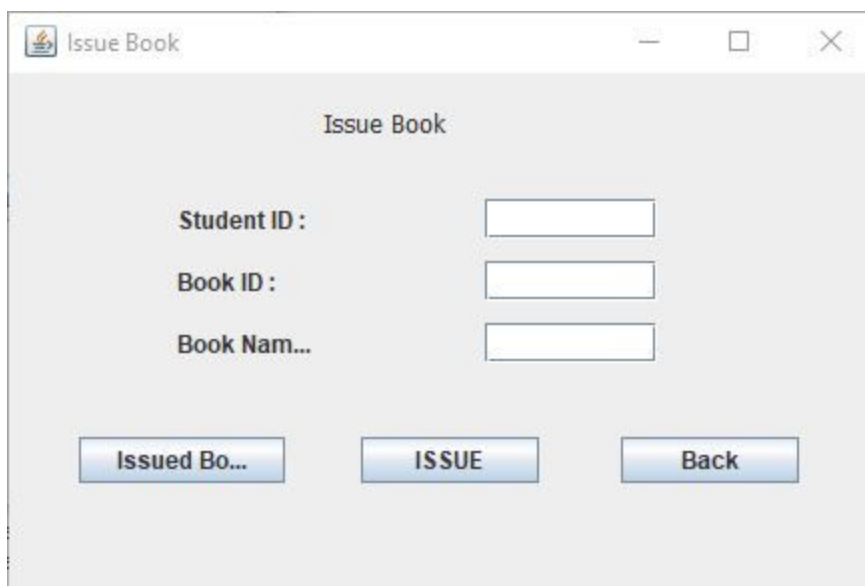
```

Swing (GUI Builder) classes:

- HomePage



- Issue



```

java  Issue.java  List.java  Return.java  Node.java

JButton btnNewButton = new JButton("ISSUE");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        if (Istudentid.getText().isEmpty() || Ibookid.getText().isEmpty() || Ibookname.getText().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Please provide the details");
        }
        else{
            String std = Istudentid.getText();
            String bi = Ibookid.getText();
            String bn = Ibookname.getText();
            boolean res=l.insert(std,bi,bn);
            if (res==true) {
                count++;
                JOptionPane.showMessageDialog(null, "Book Issued");
                Istudentid.setText(null);
                Ibookid.setText(null);
                Ibookname.setText(null);
            }
            else {
                JOptionPane.showMessageDialog(null, "Book not Issued");
                Istudentid.setText(null);
                Ibookid.setText(null);
                Ibookname.setText(null);}
        }
    }
}

```

```

java  Issue.java  List.java  Return.java  Node.java

});
btnNewButton.setBounds(176, 182, 89, 23);
frmIssueBook.getContentPane().add(btnNewButton);

JButton btnNewButton_1 = new JButton("Back");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        HomePage window = new HomePage();
        window.setVisible(true);
    }
});
btnNewButton_1.setBounds(306, 182, 89, 23);
frmIssueBook.getContentPane().add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("Issued Books");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        l.printlist();
    }
});
btnNewButton_2.setBounds(35, 182, 103, 23);
frmIssueBook.getContentPane().add(btnNewButton_2);

frmIssueBook.setVisible(true);

```

- Return

Return Book

Student ID :

Book ID :

Book Na...

Fine :

```

omePage.java  Issue.java  List.java  Return.java  Node.java
5      frmReturnBook.getContentPane().add(Rfine);
6
7      JButton btnNewButton = new JButton("RETURN");
8      btnNewButton.addActionListener(new ActionListener() {
9      public void actionPerformed(ActionEvent e) {
10         if (Rstudentid.getText().isEmpty() || Rbookid.getText().isEmpty() || Rbookname.getText().isEmpty()) {
11             JOptionPane.showMessageDialog(null, "Please provide the details");
12         }
13         else {
14             String std = Rstudentid.getText();
15             String bi = Rbookid.getText();
16             String bn = Rbookname.getText();
17             Node res=l.searchWithId(bi);
18             if (res!=null) {
19                 JOptionPane.showMessageDialog(null, "Book Returned");
20             }
21             else if(res==null) {
22                 JOptionPane.showMessageDialog(null, "Book Not Issued");
23             }
24             Rstudentid.setText(null);
25             Rbookid.setText(null);
26             Rbookname.setText(null);
27         }
28     }
29 });
30 btnNewButton.setBounds(94, 216, 89, 23);
31 frmReturnBook.getContentPane().add(btnNewButton);
32
33 JButton btnBack = new JButton("Back");

```


Conclusion:

- Purpose of the above Java project was to Design and build a Java application based on Array Lists, Swing (GUI Builder).
- Arrays List was manipulated with insert and search functionalities.
- Swing application 'HomePage' designed with system function called as exit, and another gui classes has been called through this window.

Problems Encountered:

- Implementing Linked List (so switched to Array List).
- Fine functionality was not designed as described in Initial Brief.
- Searching and Deleting a specific item from Array List.