Library Management Software

**Name –** Ved Sengupta

**Class –** X B

**School –** South City International School

**Session –** 2019 - 20

**Acknowledgements**

I would like to thank our teacher, Mr. Abhishek Guha, for guiding me in making this project. He gave us the pointers to make sure that this project was a concise, yet complete one.

I would also like to thank my computer tuition teacher, Mr. Debashish Dutta, for taking the time out to review my project.

Though it is not feasible to thank anyone in particular, the Internet certainly was of great help, in checking the meaning of errors, as well as discovering new ways of approaching the same problem.

Thank You!

**Index**

1. **Objective of Project (Page 3)**
2. **Hardware Used (Page 3)**
3. **Software Used (Page 3)**
4. **Operation Manual (Page 3)**
5. **Functions used (Page 4)**
6. **Coding (Page 5 – 42)**
7. **Output (Page 43 – 50)**
8. **Bibliography (Page 51)**

**Objective:**

The main objective of the project is to make it easy for a library to store the records of its members, including their ID, name, address and phone number, as well as the huge inventory of books, including their accession number, title, author, price and quantity. The librarian can then add, remove, and search for records of books and members, as well as track the books which have been borrowed using a separate list.

**Hardware Used:**

* **Processor –** Intel Core m5
* **RAM –** 4 GB

**Software Used:**

* **Platform –** Windows 10
* **Front End –** Java 8
* **IDE –** BlueJ
* **Back End –** Notepad

**Operation Manual:**

1. Copy the code given below into three separate documents (Class Names – Library2, Member2, Book2)
2. Save the documents as java files
3. Open Command Prompt
4. Compile ‘Library.java’ using the ‘javac’ function
5. Execute the file using the ‘java’ function

**Functions Used:**

**Coding:**

**Class – Library2 –**

import java.util.\*;

public class Library2

{

public static void main(String args[])

{

System.out.println("Welcome to the Library Management Software!");

Library2 get = new Library2();

get.displayMenu();

}

public void displayMenu()

{

Scanner key = new Scanner(System.in);

Member2 link1 = new Member2();

Book2 link2 = new Book2();

int choice, backCount = 0, error = 0;//backCount - Variable for going back to previous menu

do

{

backCount = 0;

System.out.printf("\n1)Members \n2)Books \n3)EXIT \n");

System.out.print("Choose your option number: ");

choice = key.nextInt();

System.out.print("\u000C");//Clears up terminal window

switch(choice)

{

case 1:

System.out.printf("\n1.Display all records \n2.Insert a record \n3.Delete a record \n4.Search a record \n5.Exit \n");

System.out.print("Choose your option number: ");

int choice1 = key.nextInt();

System.out.print("\u000C");

switch(choice1)

{

case 1:

link1.display();

backCount = link1.postMenu();

System.out.print("\u000C");

break;

case 2:

error = link1.create();

if(error != 1)//Exits out of program if there are any errors

{

backCount = link1.postMenu();

System.out.print("\u000C");

}

break;

case 3:

backCount = link1.deleteRecord();

link1.copyBack();

if(backCount != 1 && backCount != 0)

{

backCount = link1.postMenu();

}

System.out.print("\u000C");

break;

case 4:

backCount = link1.searchRecord();

if(backCount != 1 && backCount != 0)

{

backCount = link1.postMenu();

}

System.out.print("\u000C");

break;

case 5:

backCount = 0;

break;

default:

System.out.printf("\nSorry! Wrong option. Restart the program and try again");

}

break;

case 2:

System.out.printf("\n1.Display all books \n2.Add Book \n3.Remove Book \n4.Search Book(s) \n5.Borrow a book \n6.Return a book \n7.Borrowed Books \n8.Exit \n");

System.out.print("Choose your option number: ");

choice1 = key.nextInt();

System.out.print("\u000C");

switch(choice1)

{

case 1:

link2.display();

backCount = link2.postMenu();

System.out.print("\u000C");

break;

case 2:

error = link2.create();

if(error != 1)

{

backCount = link2.postMenu();

System.out.print("\u000C");

}

break;

case 3:

backCount = link2.deleteRecord();

link2.copyBack(0);

if(backCount != 1 && backCount != 0)

{

backCount = link2.postMenu();

}

System.out.print("\u000C");

break;

case 4:

backCount = link2.searchRecord();

if(backCount != 1 && backCount != 0)

{

backCount = link2.postMenu();

}

System.out.print("\u000C");

break;

case 5:

link2.borrowBook();

backCount = link2.postMenu();

System.out.println("\u000C");

break;

case 6:

link2.returnBook();

link2.copyBack(1);

backCount = link2.postMenu();

System.out.println("\u000C");

break;

case 7:

link2.displayBorrowedBooks();

backCount = link2.postMenu();

System.out.println("\u000C");

break;

case 8:

backCount = 0;

break;

default:

System.out.printf("\nSorry! Wrong option. Restart the program and try again");

}

break;

case 3:

backCount = 0;

break;

default:

System.out.printf("\nSorry! Wrong option. Restart the program and try again");

}

}while(backCount == 1);

System.out.println("Thank you!\nClick on the cross at the top-right corner to exit");

}

}

**Class – Member2 –**

import java.util.\*;

import java.io.\*;

public class Member2

{

Scanner key1 = new Scanner(System.in);

Scanner key2 = new Scanner(System.in);

int backCount = 0, copied = 0;

public int checkNum()

{

int personCount = 0;

try

{

Scanner rd = new Scanner(new File("member2.txt"));

while(rd.hasNext())

{

if(rd.hasNext("MEMBER"))

{

rd.next();

personCount = rd.nextInt();

}

else

{

rd.next();

}

}

rd.close();

}

catch(Exception e)

{

System.out.println("File not created yet");

}

return personCount;

}

public int create()

{

int error = 0, count1 = 0, count2 = 0;

try

{

while(count1 == 0)

{

count1 = 0;

count2 = 0;

FileWriter fw = new FileWriter("member2.txt", true);

Formatter fm = new Formatter(fw);

int personCount = checkNum();

System.out.println("MEMBER "+(personCount+1)+"\r\n");

fm.format("\r\nMEMBER %d \r\n\r\n", (personCount+1));

System.out.print("ID: ");

int mem\_ID = key1.nextInt();

fm.format("ID: %d \r\n", mem\_ID);

System.out.print("Name: ");

String mem\_name = key2.nextLine();

fm.format("Name: %s \r\n", mem\_name);

System.out.print("Address: ");

String mem\_add = key2.nextLine();

fm.format("Address: %s \r\n", mem\_add);

System.out.print("Phone Number: ");

String mem\_phone = key2.nextLine();

fm.format("Mobile Number: %s \r\n", mem\_phone);

fm.close();

fw.close();

System.out.print("\nDo you want to insert another record? Yes/No? \nChoose: ");

while(count2 == 0)

{

String ch = key2.nextLine();

if(ch.equalsIgnoreCase("Yes"))

{

count1 = 0;

count2 = 1;

}

else if(ch.equalsIgnoreCase("No"))

{

count1 = 1;

count2 = 1;

}

else

{

System.out.print("Try again. 'Yes' or 'No'. \nChoose: ");

}

}

}

}

catch(Exception e)

{

System.out.println("There was an error. Restart the program");

error = 1;

}

return error;

}

public void display()

{

try

{

Scanner sf = new Scanner(new File("member2.txt"));

while(sf.hasNext())

{

System.out.println(sf.nextLine());

}

sf.close();

}

catch(Exception e)

{

System.out.println("No records here");

}

}

public int postMenu()

{

backCount = 0;

System.out.print("\n1.Back to main menu \n2.Exit \nChoose: ");

int count1 = key1.nextInt();

switch(count1)

{

case 1:

backCount = 1;

break;

case 2:

backCount = 0;

break;

default:

System.out.println("\nWrong option");

}

return backCount;

}

public int searchRecord()

{

backCount = -1;

int count;

System.out.print("\n1.Search by ID \n2.Search by name \n3.Back to main menu \n4.Exit \nChoose your option number: ");

int choice = key1.nextInt();

try

{

Scanner read2 = new Scanner(new File("member2.txt"));

Scanner read1 = new Scanner(new File("member2.txt"));

int pCount = 0, check = 0;

switch(choice)

{

case 1:

System.out.print("\nEnter ID: ");

int id = key1.nextInt();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("MEMBER"))

{

read1.next();

read2.next();

pCount = read1.nextInt();

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("ID:"))

{

read1.next();

if(read1.nextInt() == id)

{

System.out.println("\r\nMEMBER "+pCount);

read2.nextLine();

while(read2.hasNext("MEMBER") == false && read2.hasNextLine())

{

System.out.println(read2.nextLine());

}

check = 1;

}

else

{

for(count = 1; count<=6; count++)

{

read2.nextLine();

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

break;

case 2:

System.out.print("Enter name: ");

String name = key2.nextLine();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("MEMBER"))

{

read1.next();

read2.next();

pCount = read1.nextInt();

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Name:"))

{

read1.next();

if(read1.nextLine().equalsIgnoreCase(" "+name+" "))

{

System.out.println("\r\nMEMBER "+pCount);

read2.nextLine();

while(read2.hasNext("MEMBER") == false && read2.hasNextLine())

{

System.out.println(read2.nextLine());

}

check = 1;

}

else

{

for(count = 1; count<=6; count++)

{

read2.nextLine();

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

break;

case 3:

backCount = 1;

break;

case 4:

backCount = 0;

break;

default:

System.out.println("Wrong option");

}

read1.close();

read2.close();

}

catch(Exception e)

{

System.out.println("\nNothing here");

}

return backCount;

}

public int deleteRecord()

{

backCount = -1;

copied = 0;

int personCount = 0, count, done = 0, check = 0;

System.out.print("\n1.Delete by ID \n2.Delete by name \n3.Back to main menu \n4.Exit \nChoose your option number: ");

int choice = key1.nextInt();

try

{

Scanner read2 = new Scanner(new File("member2.txt"));

Scanner read1 = new Scanner(new File("member2.txt"));

FileWriter file = new FileWriter("copyMem2.txt", false);

Formatter wr = new Formatter(file);

switch(choice)

{

case 1:

System.out.print("\nEnter ID: ");

int id = key1.nextInt();

System.out.println();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("MEMBER"))

{

read1.next();

read2.next();

if(done == 0)

{

personCount = read1.nextInt();

}

else

{

personCount = read1.nextInt() - 1;

}

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("ID:"))

{

read1.next();

if(read1.nextInt() == id)

{

for(count = 1; count<=6; count++)

{

read2.nextLine();

}

done = 1;

check = 1;

}

else

{

wr.format("\r\nMEMBER %d\r\n", personCount);

read2.nextLine();

while(read2.hasNext("MEMBER") == false && read2.hasNextLine())

{

wr.format("%s\r\n", read2.nextLine());

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

else

{

System.out.println("Record deleted successfully");

}

copied = 1;

break;

case 2:

System.out.print("Enter name: ");

String name = key2.nextLine();

System.out.println();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("MEMBER"))

{

read1.next();

read2.next();

if(done == 0)

{

personCount = read1.nextInt();

}

else

{

personCount = read1.nextInt() - 1;

}

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Name:"))

{

read1.next();

if(read1.nextLine().equalsIgnoreCase(" "+name+" "))

{

for(count = 1; count<=6; count++)

{

read2.nextLine();

}

done = 1;

check = 1;

}

else

{

wr.format("\r\nMEMBER %d\r\n", personCount);

read2.nextLine();

while(read2.hasNext("MEMBER") == false && read2.hasNextLine())

{

wr.format("%s\r\n", read2.nextLine());

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

else

{

System.out.println("Record deleted successfully");

}

copied = 1;

break;

case 3:

backCount = 1;

break;

case 4:

backCount = 0;

break;

default:

System.out.println("Wrong option");

}

read1.close();

read2.close();

wr.close();

file.close();

}

catch(Exception e)

{

System.out.println("Nothing here");

}

return backCount;

}

public void copyBack()

{

try

{

if(copied == 1)

{

Scanner rd = new Scanner(new File("copyMem2.txt"));

FileWriter file = new FileWriter("member2.txt", false);

Formatter wr = new Formatter(file);

while(rd.hasNext())

{

wr.format("%s\r\n", rd.nextLine());

}

rd.close();

wr.close();

file.close();

}

}

catch(Exception e)

{

System.out.println("Nothing here");

}

}

}

**Class – Book2 –**

import java.util.\*;

import java.io.\*;

public class Book2

{

Scanner key1 = new Scanner(System.in);

Scanner key2 = new Scanner(System.in);

int backCount = 0, copied = 0;

public int checkNum()

{

int bookCount = 0;

try

{

Scanner rd = new Scanner(new File("book2.txt"));

while(rd.hasNext())

{

if(rd.hasNext("BOOK"))

{

rd.next();

bookCount = rd.nextInt();

}

else

{

rd.next();

}

}

rd.close();

}

catch(Exception e)

{

System.out.println("Nothing here");

}

return bookCount;

}

public int create()

{

int error = 0, count1 = 0, count2 = 0;

try

{

while(count1 == 0)

{

count1 = 0;

count2 = 0;

FileWriter fw = new FileWriter("book2.txt", true);

Formatter fm = new Formatter(fw);

int bookCount = checkNum();

System.out.println("BOOK "+(bookCount+1)+"\r\n");

fm.format("\r\nBOOK %d \r\n\r\n", (bookCount+1));

System.out.print("Accession\_Number: ");

int book\_acc\_num = key1.nextInt();

fm.format("Accession\_Number: %d \r\n", book\_acc\_num);

System.out.print("Title: ");

String book\_title = key2.nextLine();

fm.format("Title: %s \r\n", book\_title);

System.out.print("Author: ");

String book\_author = key2.nextLine();

fm.format("Author: %s \r\n", book\_author);

System.out.print("Price: Rs.");

float book\_price = key1.nextFloat();

fm.format("Price: Rs.%f \r\n", book\_price);

System.out.print("Quantity: ");

int book\_quan = key1.nextInt();

fm.format("Quantity: %d \r\n", book\_quan);

fm.close();

fw.close();

System.out.print("\nDo you want to insert another record? Yes/No? \nChoose: ");

while(count2 == 0)

{

String ch = key2.nextLine();

if(ch.equalsIgnoreCase("Yes"))

{

count1 = 0;

count2 = 1;

}

else if(ch.equalsIgnoreCase("No"))

{

count1 = 1;

count2 = 1;

}

else

{

System.out.print("Try again. 'Yes' or 'No'. \nChoose: ");

}

}

}

}

catch(Exception e)

{

System.out.println("There was an error. Restart the program");

error = 1;

}

return error;

}

public void display()

{

try

{

Scanner sf = new Scanner(new File("book2.txt"));

while(sf.hasNext())

{

System.out.println(sf.nextLine());

}

sf.close();

}

catch(Exception e)

{

System.out.println("Nothing here");

}

}

public int postMenu()

{

backCount = 0;

System.out.print("\n1.Back to main menu \n2.Exit \nChoose: ");

int count1 = key1.nextInt();

switch(count1)

{

case 1:

backCount = 1;

break;

case 2:

backCount = 0;

break;

default:

System.out.println("\nWrong option");

}

return backCount;

}

public int searchRecord()

{

backCount = -1;

int count;

System.out.print("\n1.Search by Accession Number \n2.Search by title \n3.Search by author \n4.Back to main menu \n5.Exit \nChoose your option number: ");

int choice = key1.nextInt();

try

{

Scanner read2 = new Scanner(new File("book2.txt"));

Scanner read1 = new Scanner(new File("book2.txt"));

int pCount = 0, check = 0;

switch(choice)

{

case 1:

System.out.print("\nEnter Accession Number: ");

int id = key1.nextInt();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("BOOK"))

{

read1.next();

read2.next();

pCount = read1.nextInt();

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Accession\_Number:"))

{

read1.next();

if(read1.nextInt() == id)

{

System.out.println("\r\nBOOK "+pCount);

read2.nextLine();

while(read2.hasNext("BOOK") == false)

{

System.out.println(read2.nextLine());

}

check = 1;

}

else

{

for(count = 1; count<=7; count++)

{

read2.nextLine();

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

break;

case 2:

System.out.print("Enter title: ");

String name = key2.nextLine();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("BOOK"))

{

read1.next();

read2.next();

pCount = read1.nextInt();

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Title:"))

{

read1.next();

if(read1.nextLine().equalsIgnoreCase(" "+name+" "))

{

System.out.println("\r\nBOOK "+pCount);

read2.nextLine();

while(read2.hasNext("BOOK") == false)

{

System.out.println(read2.nextLine());

}

check = 1;

}

else

{

for(count = 1; count<=7; count++)

{

read2.nextLine();

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

break;

case 3:

System.out.print("Enter author: ");

String auth = key2.nextLine();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("BOOK"))

{

read1.next();

read2.next();

pCount = read1.nextInt();

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Author:"))

{

read1.next();

if(read1.nextLine().equalsIgnoreCase(" "+auth+" "))

{

System.out.println("\r\nBOOK "+pCount);

read2.nextLine();

while(read2.hasNext("BOOK") == false)

{

System.out.println(read2.nextLine());

}

check = 1;

}

else

{

for(count = 1; count<=7; count++)

{

read2.nextLine();

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

break;

case 4:

backCount = 1;

break;

case 5:

backCount = 0;

break;

default:

System.out.println("Wrong option");

}

read1.close();

read2.close();

}

catch(Exception e)

{

}

return backCount;

}

public int deleteRecord()

{

backCount = -1;

copied = 0;

int personCount = 0, count, done = 0, check = 0;

System.out.print("\n1.Delete by Accession Number \n2.Delete by title \n3.Back to main menu \n4.Exit \nChoose your option number: ");

int choice = key1.nextInt();

try

{

Scanner read2 = new Scanner(new File("book2.txt"));

Scanner read1 = new Scanner(new File("book2.txt"));

FileWriter file = new FileWriter("copyBook2.txt", false);

Formatter wr = new Formatter(file);

switch(choice)

{

case 1:

System.out.print("\nEnter Accession Number: ");

int id = key1.nextInt();

System.out.println();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("BOOK"))

{

read1.next();

read2.next();

if(done == 0)

{

personCount = read1.nextInt();

}

else

{

personCount = read1.nextInt() - 1;

}

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Accession\_Number:"))

{

read1.next();

if(read1.nextInt() == id)

{

for(count = 1; count<=7; count++)

{

read2.nextLine();

}

done = 1;

check = 1;

}

else

{

wr.format("\r\nBOOK %d\r\n", personCount);

read2.nextLine();

while(read2.hasNext("BOOK") == false && read2.hasNextLine())

{

wr.format("%s\r\n", read2.nextLine());

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

else

{

System.out.println("Record deleted successfully");

}

copied = 1;

break;

case 2:

System.out.print("Enter title: ");

String name = key2.nextLine();

System.out.println();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("BOOK"))

{

read1.next();

read2.next();

if(done == 0)

{

personCount = read1.nextInt();

}

else

{

personCount = read1.nextInt() - 1;

}

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Title:"))

{

read1.next();

if(read1.nextLine().equalsIgnoreCase(" "+name+" "))

{

for(count = 1; count<=7; count++)

{

read2.nextLine();

}

done = 1;

check = 1;

}

else

{

wr.format("\r\nBOOK %d\r\n", personCount);

read2.nextLine();

while(read2.hasNext("BOOK") == false && read2.hasNextLine())

{

wr.format("%s\r\n", read2.nextLine());

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

else

{

System.out.println("Record deleted successfully");

}

copied = 1;

break;

case 3:

backCount = 1;

break;

case 4:

backCount = 0;

break;

default:

System.out.println("Wrong option");

}

read1.close();

read2.close();

wr.close();

file.close();

}

catch(Exception e)

{

}

return backCount;

}

public void copyBack(int poly)

{

try

{

Scanner rd = new Scanner(new File("copyBook2.txt"));

FileWriter file;

if(poly == 1)

{

file = new FileWriter("bookBorrow2.txt", false);

Formatter wr = new Formatter(file);

while(rd.hasNext())

{

wr.format("%s\r\n", rd.nextLine());

}

wr.close();

file.close();

}

else if(copied == 1)

{

file = new FileWriter("book2.txt", false);

Formatter wr = new Formatter(file);

while(rd.hasNext())

{

wr.format("%s\r\n", rd.nextLine());

}

wr.close();

file.close();

}

rd.close();

}

catch(Exception e)

{

System.out.println("Nothing here");

}

}

public void borrowBook()

{

backCount = 0;

int count;

try

{

Scanner read2 = new Scanner(new File("book2.txt"));

Scanner read1 = new Scanner(new File("book2.txt"));

FileWriter file = new FileWriter("bookBorrow2.txt", true);

Formatter wr = new Formatter(file);

int pCount = 0, check = 0;

System.out.print("Enter title of the book to be borrowed: ");

String name = key2.nextLine();

System.out.println();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("BOOK"))

{

read1.next();

read2.next();

pCount = read1.nextInt();

read2.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Title:"))

{

read1.next();

if(read1.nextLine().equalsIgnoreCase(" "+name+" "))

{

wr.format("\r\nBOOK %d\r\n", pCount);

read2.nextLine();

while(read2.hasNext("BOOK") == false && read2.hasNextLine())

{

wr.format("%s\r\n", read2.nextLine());

}

check = 1;

}

else

{

for(count = 1; count<=7; count++)

{

read2.nextLine();

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

else

{

System.out.println("Kindly return the book within 7 days.");

}

read1.close();

read2.close();

wr.close();

file.close();

}

catch(Exception e)

{

System.out.println("Nothing here");

}

}

public void returnBook()

{

backCount = 0;

copied = 0;

int personCount = 0, count, done = 0, check = 0;

try

{

Scanner read2 = new Scanner(new File("bookBorrow2.txt"));

Scanner read1 = new Scanner(new File("bookBorrow2.txt"));

FileWriter file = new FileWriter("copybook2.txt", false);

Formatter wr = new Formatter(file);

System.out.print("Enter title of the book to be returned: ");

String name = key2.nextLine();

System.out.println();

while(read1.hasNext() && read2.hasNext())

{

if(read1.hasNext("BOOK"))

{

read1.next();

read2.next();

personCount = read1.nextInt();

}

else

{

read1.nextLine();

//read2.next();

}

if(read1.hasNext("Title:"))

{

read1.next();

if(read1.nextLine().equalsIgnoreCase(" "+name+" "))

{

for(count = 1; count<=7; count++)

{

read2.nextLine();

}

check = 1;

}

else

{

wr.format("\r\nBOOK %d\r\n", personCount);

read2.nextLine();

while(read2.hasNext("BOOK") == false && read2.hasNextLine())

{

wr.format("%s\r\n", read2.nextLine());

}

}

}

}

if(check == 0)

{

System.out.println("No results found");

}

else

{

System.out.println("Thank you!");

}

read1.close();

read2.close();

wr.close();

file.close();

}

catch(Exception e)

{

}

}

public void displayBorrowedBooks()

{

try

{

Scanner sf = new Scanner(new File("bookBorrow2.txt"));

while(sf.hasNext())

{

System.out.println(sf.nextLine());

}

sf.close();

}

catch(Exception e)

{

System.out.println("Nothing here");

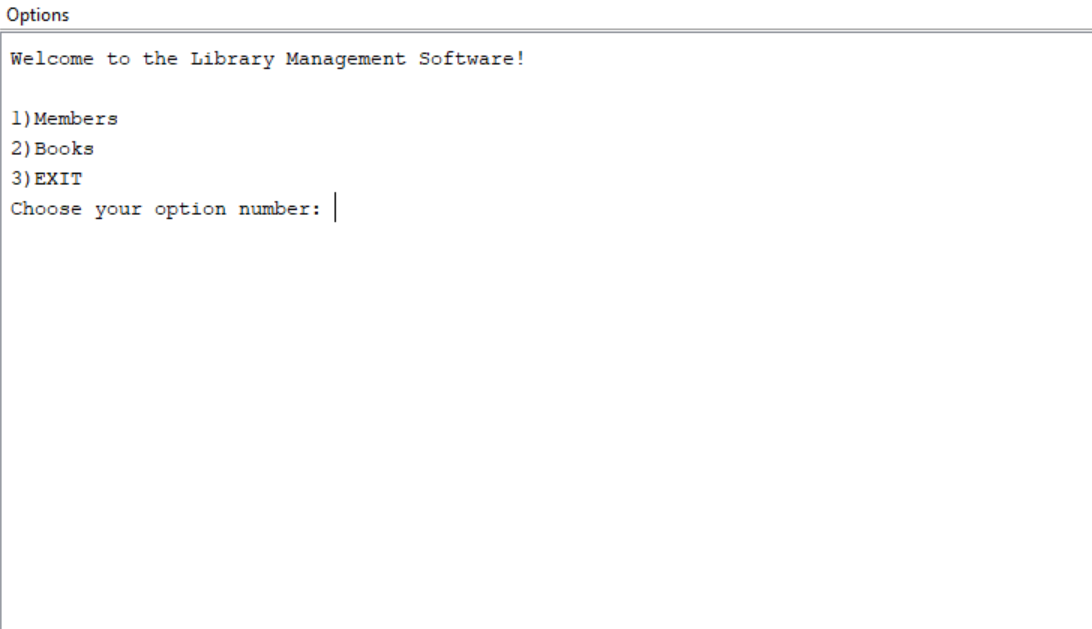
}

}

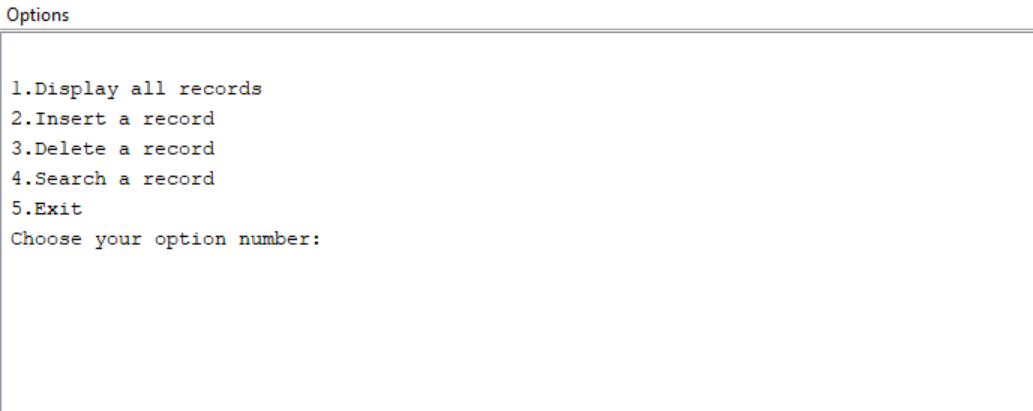
}

**Output:**

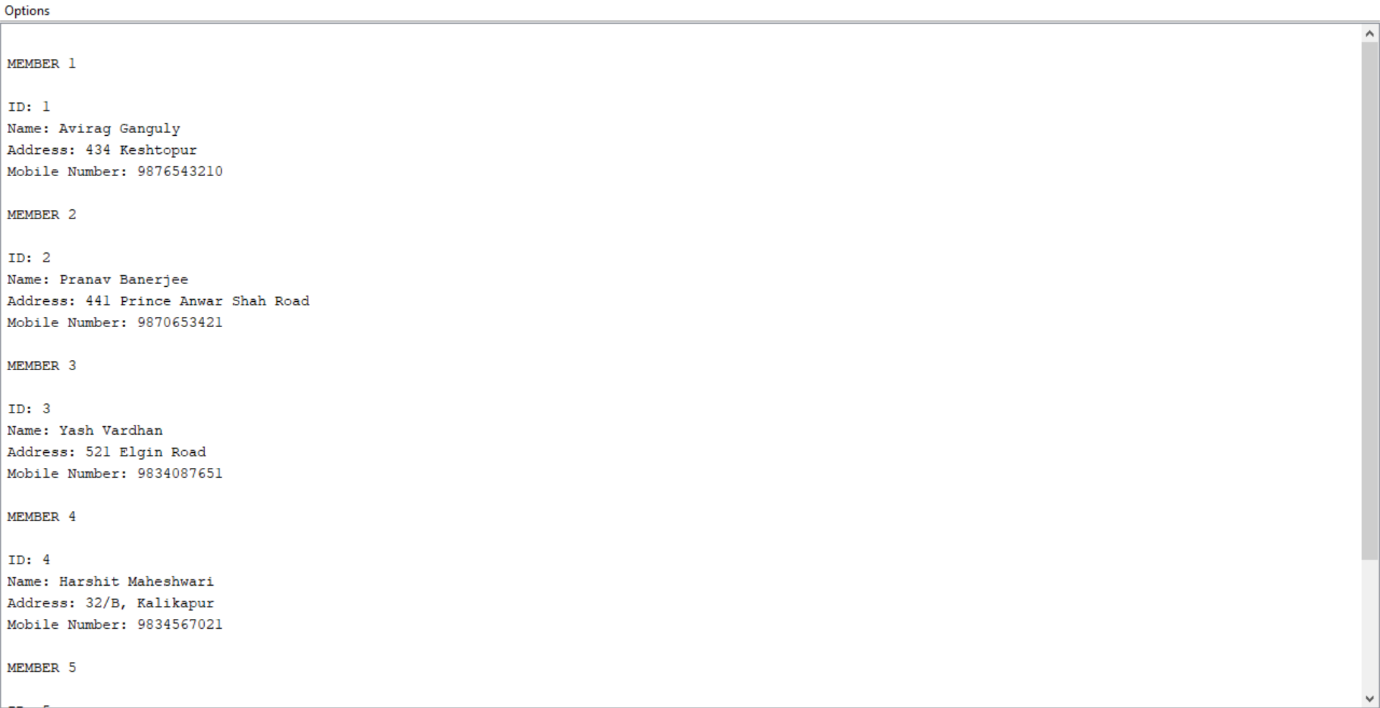
Main Menu:



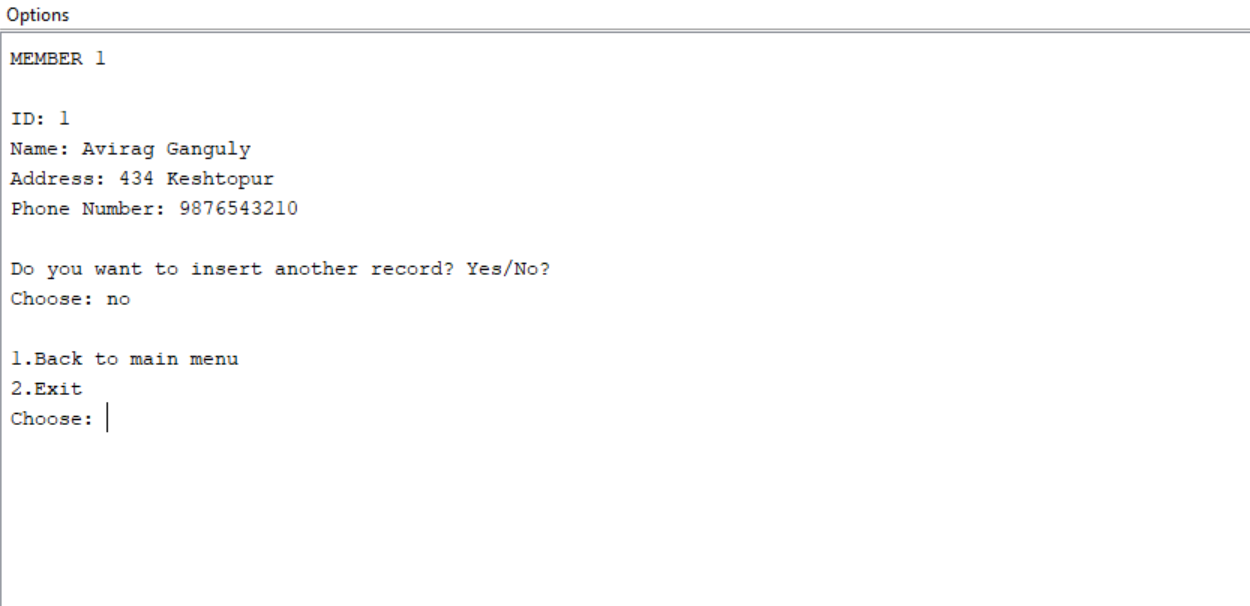
1. Members:



1. Display all records:



1. Insert Record:



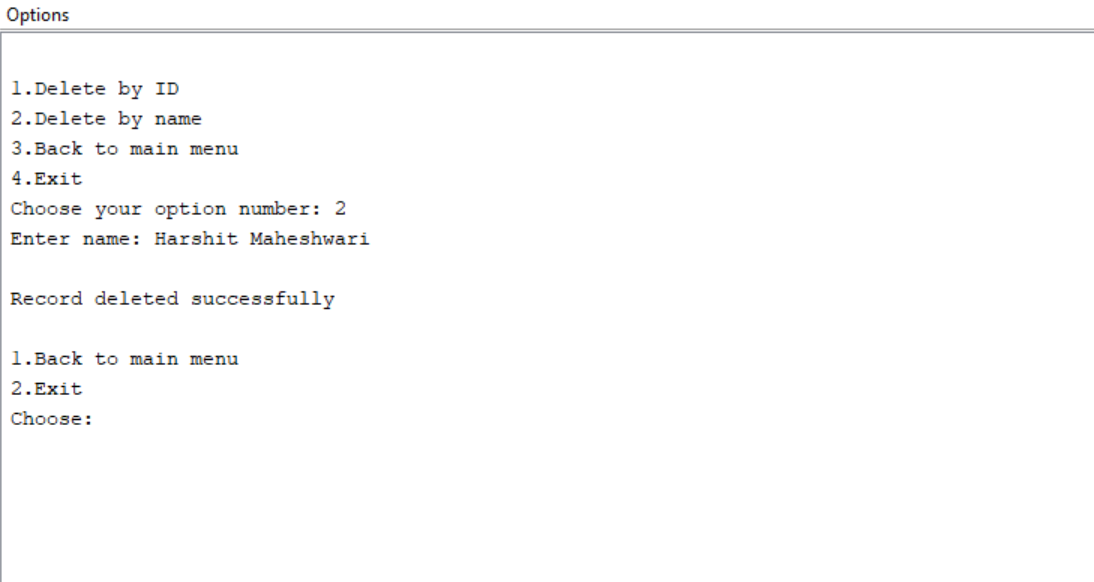
1. Delete Record:

Option 1 – By ID



1. Delete Record:

Option 2 – By name



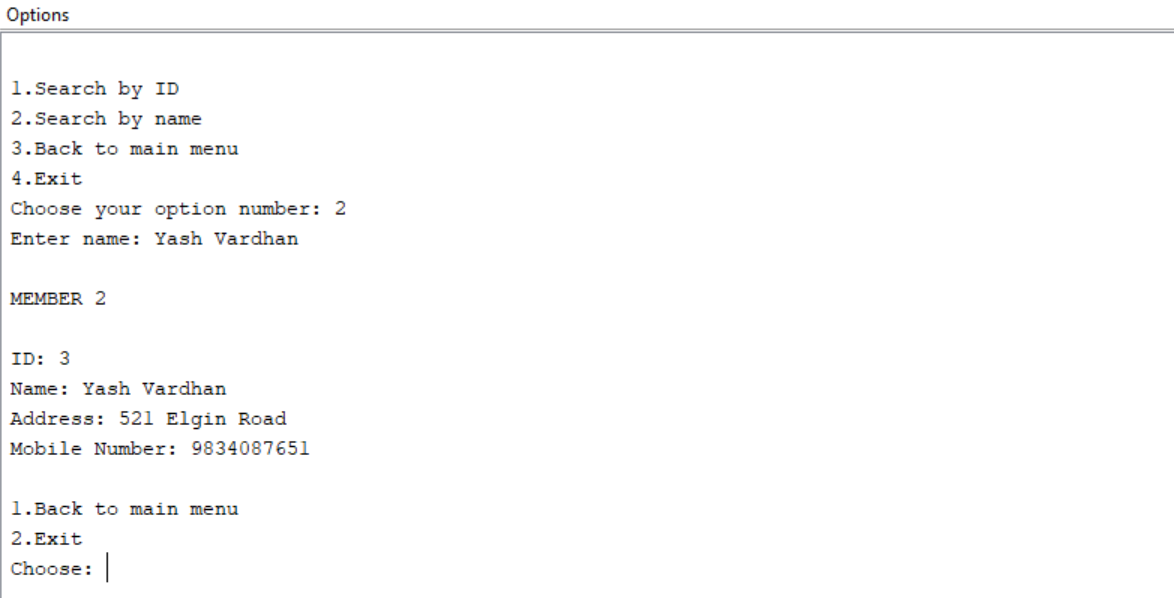
1. Search Record:

Option 1 – By ID

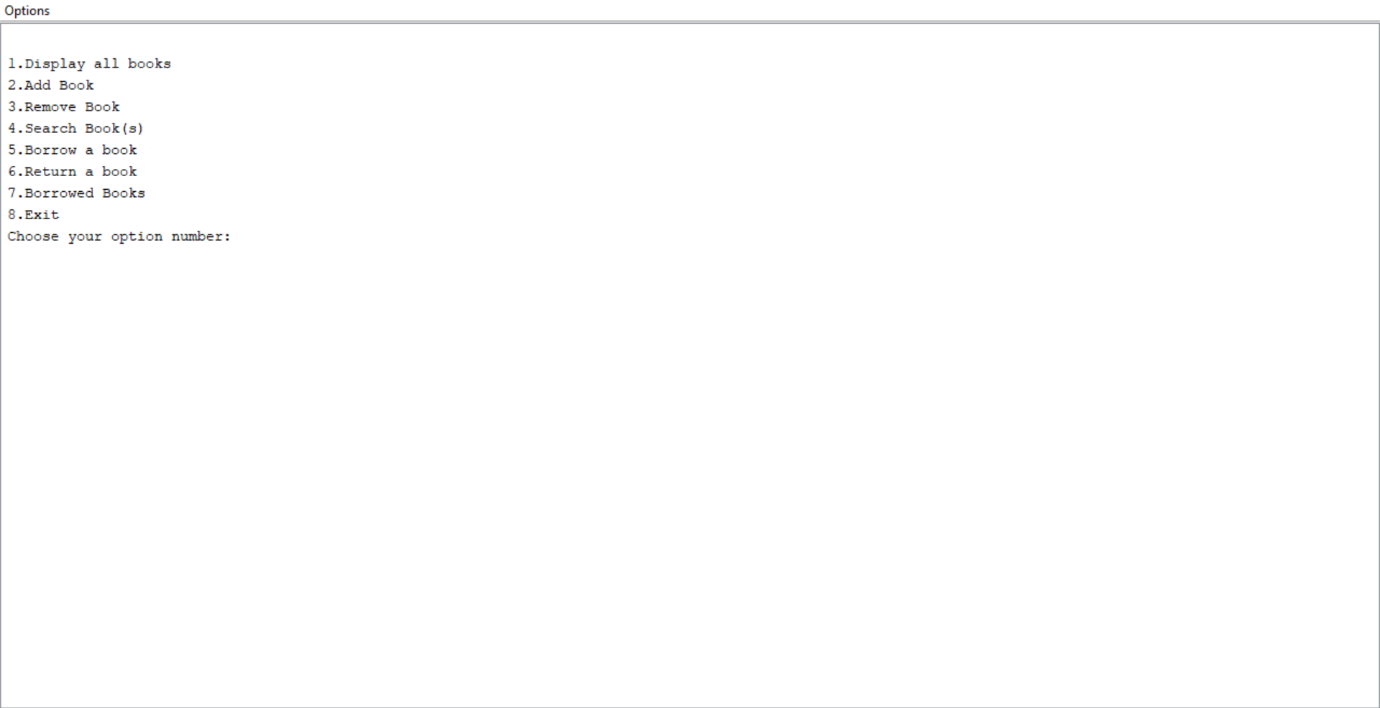


1. Search Record:

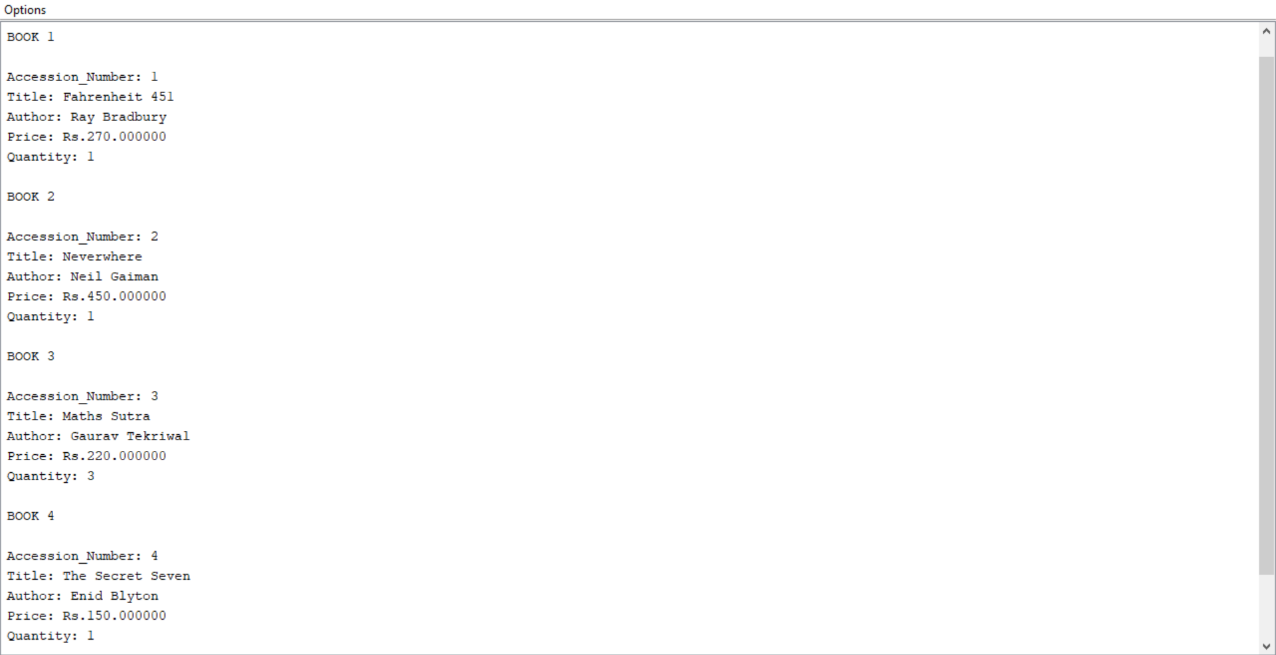
Option 2 – By name



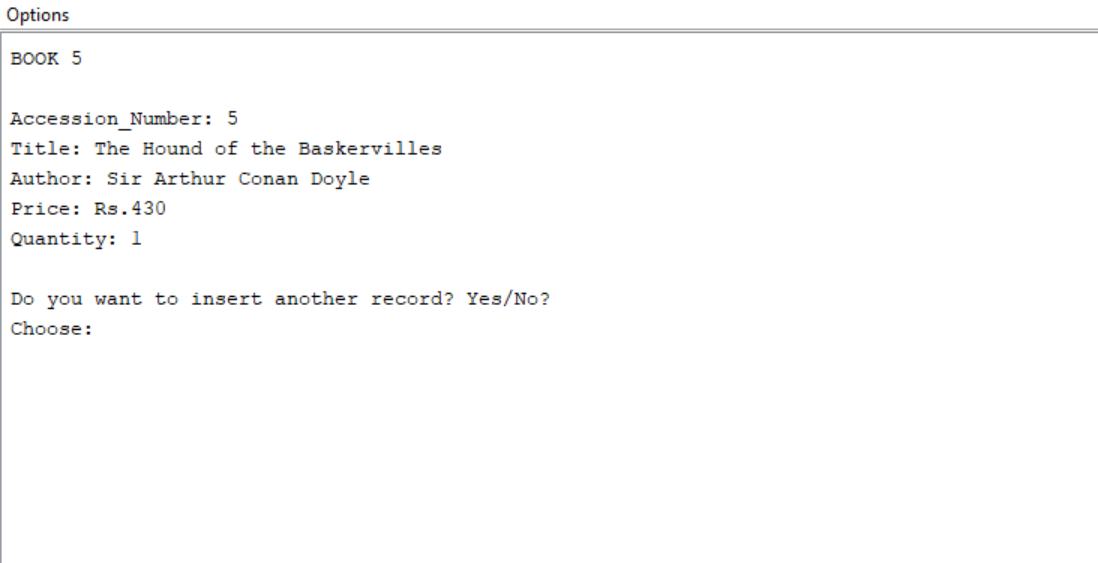
1. Books:



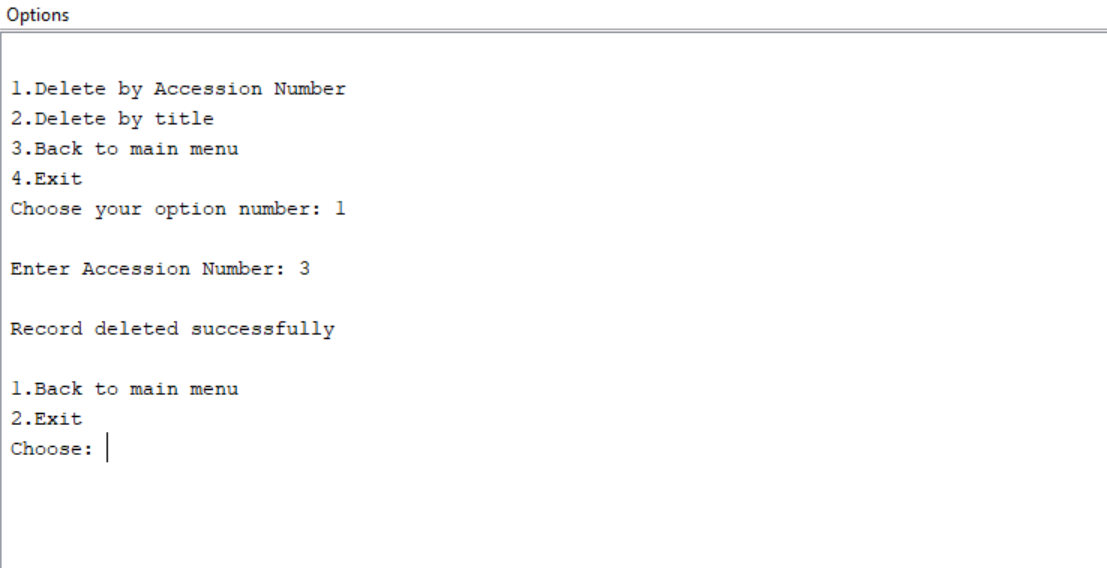
1. Display all books:



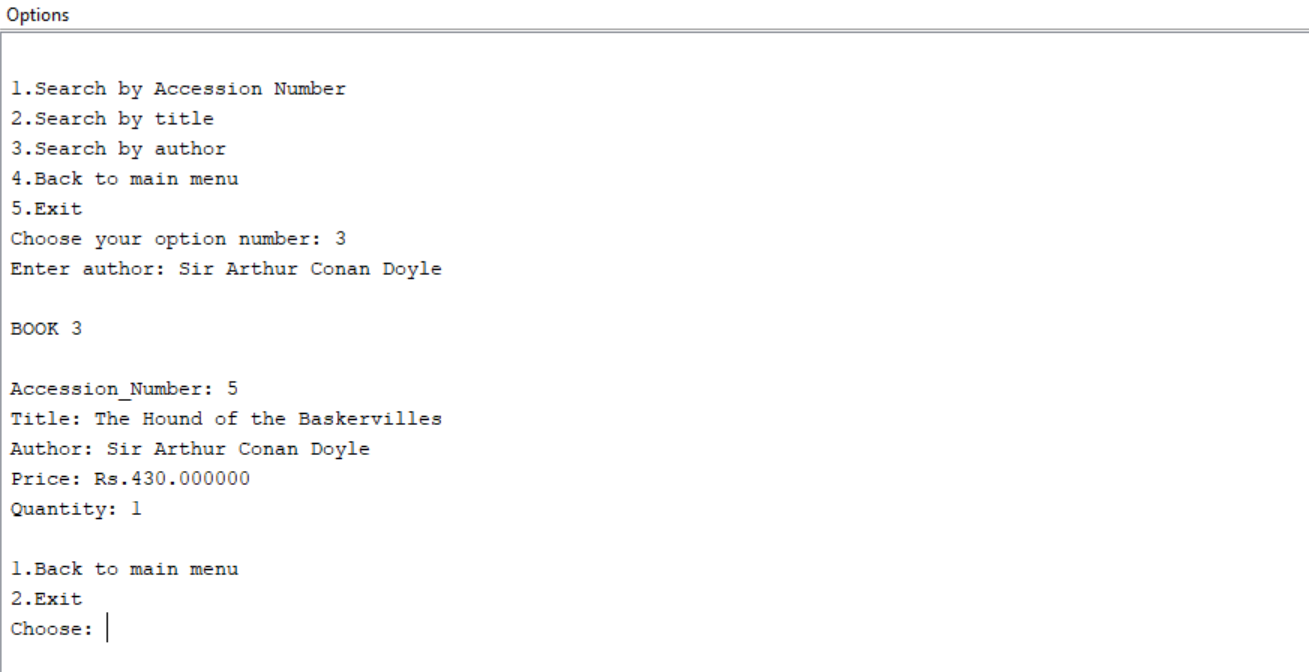
1. Add Book:



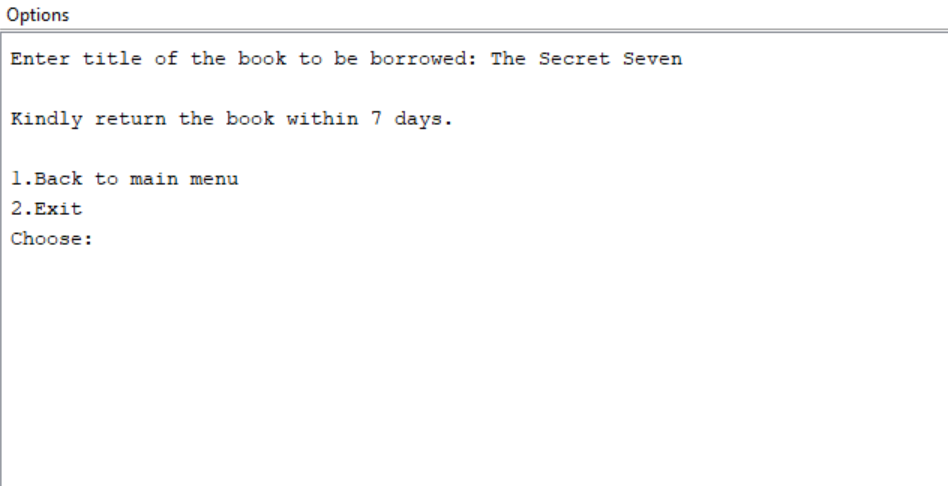
1. Remove Book:



1. Search Book:  
    Option 3 – Author



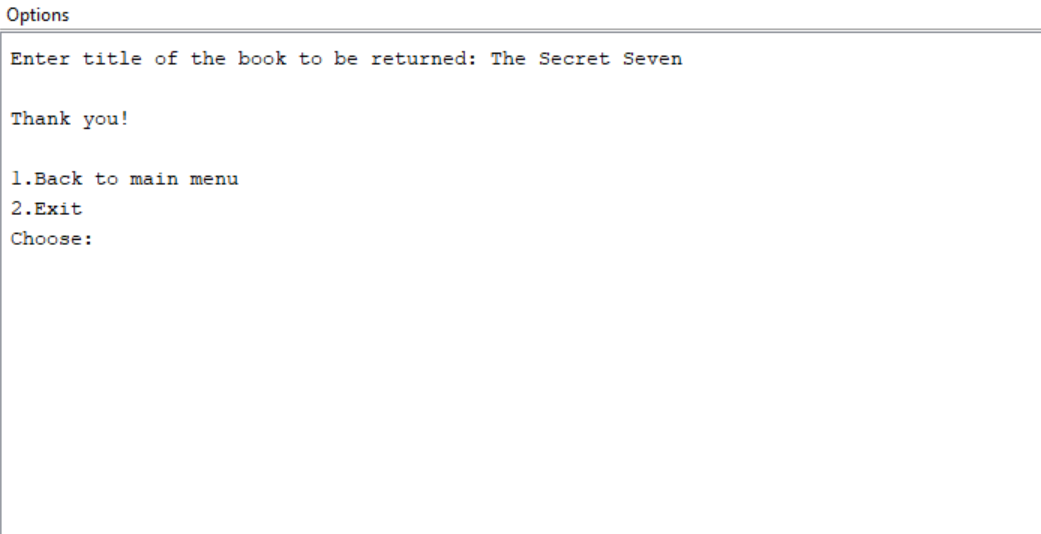
1. Borrow Book:



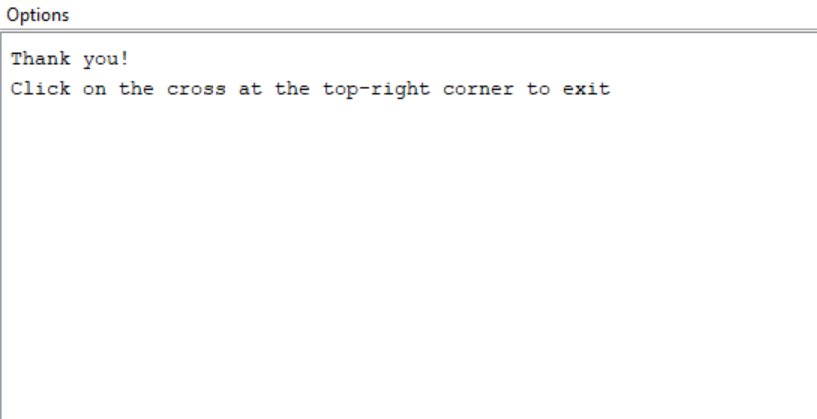
1. Display Borrowed Books:



1. Return Book:



1. EXIT:



Note: All records have been stored in .txt files using Notepad.

**Bibliography:**

* <https://www.tutorialspoint.com/>
* <https://geeksforgeeks.org/>
* <https://stackoverflow.com/questions/>
* <https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>