Sangey Lama
Mac 190-8141
Research Project report and program
Library Management System

1. Problem Description

Library Management System is the program that I chose to code to enhance and hone my coding skills, but most importantly to have a grasp on program that is far from basics and have experience on program that is applied on next level. This not only helps me advance my coding skills but to also truly understand and go deep in to the topic that I am investigating, library system, for further knowledge. Practicing and implementing the library system program has and will aid me to create less logic error which I often make while coding, and memorize the syntaxes that was used in the coding. I also learned things myself while working and spending time on this project. Since I made lot of mistakes, all the error and the red underlines really bothered me, so implementing a maintainable code is one of the main part of learning in programming and its more crucial than the code running perfectly without a problem.

A Library Management System which is also called as Integrated library system, is a software that is programmed to manage the basic activity and the functions of a library. Library System is one of the valuable resource planning system for a library, allowing the user to track all the recorded unit, owned items, saved accounts and money supplied to each item. This helps the library to become well organized and lets the software to become a profound solution. If there weren't a library management system software then it will a big mess for the librarians and for users. All the library item and account recorded on a paper would only lead to waste of paper and off course feel very disorganized. There is likely a high chance of losing an old recorded file due to overload of paper everywhere. Lastly, one crucial thing is that there is a loss of time and cost since dealing with paper absorb lot of time and money. Hence, taking the digital pathway is an excellent and a convenient choice for library users. One of the important function of the LMS is that it displays details of any book recorded in the library which is available to all the user. And it also saves all the information of book issued which was attached to the library by any user. For this program to run according to its function without a problem, it is vital for the Library System to maintain data on all the recorded books. It should store databases for books which stores information regarding a book author, title, price, and the ISBN number.

2. Introduction

The library system that I programmed includes four built in library functions for the user to decide. Since there are four options, the program will ask the user to input and choose from one of the four choices. The options consist of attaching books to the library, revealing the recorded books by user, save and exit, and load a library file. The first option allows the user the add the book attributes inside the library. The book attributes include title, author, ISBN number and the price for the book. After the user enters all the information for the book, the data gets saved and repeatedly asks for another option within the four choices. If the users go for the first option again then they can append another book which is stored in to the data. The second option displays and lists all the books listed by the user. For Instance, if the user used the first option to add a book called "The Mocking Bird" and "Life of pie", then selecting the second option would display and list both recorded book. In the third option, user can save the data of the book in to a file and exit out of the program. The user has alternative on what to name their file and the program will say, "Your file has been saved", if the file gets saved successfully. The file will be saved inside the same project where the program is running. After the file is saved the program exits. It is possible to save a new listed book in to an old saved file by inputting the same file name again. In the fourth option, saved library file can be loaded and opened in the program by entering the file name. If the user inputs the wrong file name or there isn't any library file created then the program will automatically display, "No File Found." Also, If the program is used for the first time and the user wants to load a file, the user must add a book in to the data beforehand then use the third option to save a file with a name. After that user can open any saved library file as they like without a problem.

The advantages of this program are that it gets rid of additional paper work in the library, all items will be recorded in the library management system so all the files will be saved without any potential of loss, conserve time and eliminate manpower. Moreover, the program makes the system efficient and lets the user freely go through all the service provided in the program. The disadvantages are that there is a possibility that software can become corrupted with bugs, and entire data can collapse.

3. Analysis

The program did run consistently without an error after running it myself. The output displayed all the option that the user can get access to beginning from one to four. While choosing the option one, the output did ask for the book's attributes line to line and the book information could be easily typed with ease. The third feature, save and quit, option also worked well, making it easy to create a file just by enter a file name and exiting out of the program. The last option also passed the run test, I could load a file that I created before called, "My Library". Furthermore, I can display the book that I added in to that library file before.

4. Conclusion

This Library System was created using Eclipse and the structure did function properly. Utilizing this program can aid the library environment and direct many benefits to the users.

5. Code

```
//Serializable library
import java.io.Serializable;

//implements Serializable, stores a copy of the object and send it to another process
public class BookRecord implements Serializable{
	//private instances variable, book attributes
	private String NameOfTheBook, NameofWriter;
	private int isbnNumber;
	private double booksCost;

	//default constructor
	public BookRecord()
		{
			NameOfTheBook = "";
			NameofWriter = "";
			isbnNumber = 0;
			booksCost = 0;
		}

	//Constructor that takes parameter
	public BookRecord(String NameOfTheBook, String NameofWriter, int isbnNumber,
double booksCost)
	{
		this.NameOfTheBook = NameOfTheBook;
		this.NameofWriter = NameofWriter;
		this.isbnNumber = isbnNumber;
		this.booksCost = booksCost;
	}

	//this method returns a string representation of this object
	public String stringreturner()
	{
		String st = "\n";
		return st +"\nName of the book: " +NameOfTheBook+ "\nName of the writer:
" + NameofWriter+ "\nISBN Number: " + isbnNumber + "\nCost of The Book: $" +
booksCost;
	}



}




-------------------------------------------------------------------------------------




import java.io.Serializable;
import java.util.ArrayList;

//library that allows to create a list
import java.util.List;

public class MainLibrary implements Serializable{

	// this private List object will take the list from the other class BookRecord
	private List<BookRecord> bookAccumulation;
	public MainLibrary()
		{
			bookAccumulation = new ArrayList<BookRecord>();
		}


	//function method that adds the list of book
	public void appendsBook(BookRecord book)
		{
		//calling add function
			bookAccumulation.add(book);
		}

	//method that returns and displays all the books collected from the user
	public String CollectedBooks()
	{
		String display = "\n";
		// for loop that iterates through each of the book from the list
		for(int i = 0; i<bookAccumulation.size(); i++)
		{
			//object count that gets each book from the position i
			BookRecord count = bookAccumulation.get(i);
					//method from the other class "BookRecord"

			display = display+ count.stringreturner();
		}
		return display;
	}

}
```

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.Scanner;
public class PrincipleClass {
	// declaring static variable that belongs to this class itself
	static Scanner keyboard=  new Scanner(System.in);
	static Boolean executing = true;
	static String filename ="";
	static MainLibrary library = new MainLibrary();

	public static void main(String[] args)
	{
		int option;
		System.out.println("Welcome to Library Management System");
```

```
            System.out.println("In this program you can add new books to the
library, display the listed books, save a library file with the books you listed, and
load a saved library file");
            //keeps running until called false
            while(executing)
            {

                  System.out.println("\nTo attach a book to the library, Enter 1");
                  System.out.println("To display all the list of books added, Enter
2");
                  System.out.println("To Save and quit, Enter 3");
                  System.out.println("To load a library file, Enter 4");
                  System.out.print("Enter your option : ");
                  option = keyboard.nextInt();

            //if user inputs 1 then execute this, loads a library file
            switch(option)
            {
            case 4:
                  Scanner namefile = new Scanner (System.in);
                  System.out.println("To load a file, enter the name: ");
                  filename += namefile.nextLine();

                  //FileInputStream reads data from a file
                  FileInputStream read= null;

                  //reads an object from an input stream
                  ObjectInputStream objectInput =null;

                  //gives access to the file, with extension of .ser because of
serialized object
                  File file = new File(filename+".ser");
                  //executes if the file exists
                  if(file.exists())
                  {
                        try
                        {
                              read = new FileInputStream(file);
                              objectInput = new ObjectInputStream(read);
                              library = (MainLibrary) objectInput.readObject();
                              read.close();
                              objectInput.close();
                        }
                        catch(IOException e)
                        {
                              e.printStackTrace();
                        }
                        catch (ClassNotFoundException e)
                        {
                              e.printStackTrace();
                        }
                        System.out.println("Your file '"+ filename + " has been
loaded");
                  }
                  else
```

```
                    {
                           System.out.println("\nNo File found.");
                           System.out.println("Add a book then save and exit the 
file, so file could be loaded\n");
                    }
                    break;


             //if user inputs 2 then execute this, adds book to the library
          // this case allows user to add book to the library
             case 1:

                     String NameOfTheBook = "";
                     String NameofWriter= "";
                   int isbnNumber;
                double booksCost;

                Scanner space = new Scanner(System.in);
                Scanner white = new Scanner(System.in);
                System.out.print("\nEnter the name of the book: ");

                NameOfTheBook += space.nextLine();

                System.out.print("Enter the name of the writer: ");
                NameofWriter+= white.nextLine();

                System.out.print("Enter the ISBN number: ");
                isbnNumber = keyboard.nextInt();

                System.out.print("Enter the cost: ");
                booksCost = keyboard.nextDouble();

                BookRecord management= new BookRecord(
NameOfTheBook,NameofWriter,isbnNumber, booksCost);
                   library.appendsBook(management);
                   break;


             // shows all the books added to the lists
             case 2:
             System.out.println(library.CollectedBooks());
             break;


             // saves and quit
             case 3:

                    //asks the user which file to save then the program quits
                    System.out.println("You are saving the data to a file, Enter the 
file name");
                    filename = keyboard.next()+".ser";
                    executing =false;
                    //"write" file output stream for writing data to file
                    FileOutputStream write = null;
```

```
                        ObjectOutputStream output = null;
                        try {
                              write = new FileOutputStream(filename);
                              output = new ObjectOutputStream(write);
                              output.writeObject(library);
                              write.close();
                              output.close();
                        } catch (FileNotFoundException e) {
                              // TODO Auto-generated catch block
                              e.printStackTrace();
                        } catch (IOException e) {
                              // TODO Auto-generated catch block
                              e.printStackTrace();
                        }


                  }

            }
System.out.println("Your file has been saved");
}
}
```