Benjamin Knauth

CMSC 335

31 August 2014

Project #1 Description

*UMLs:*

**CMSC335\_Project\_1**

main(String args[]): void

**Library**

**Authors : ArrayList<Author>**

+Library()

+toString() : String

**Author**

-index : int

-numberBooks : int

-name : String

-address : String

**Books : ArrayLIst<Book>**

ExtraFields : ArrayList<String>

+Author()

+Author(int index, String name, String address, ArrayList<String> ExtraFields)

+toString() : String

+getIndex() : int

+setIndex(int index) : void

+getNumberBooks() : int

+setNumberBooks() : void

+getName() : String

+setName(String name) : void

+getAddress() : String

+setAddress(String address) : void

**Book**

-index : int

-authorIndex : int

-price : double

-title : String

-genre : String

-author : String

ExtraFields : ArrayList<String>

+Book()

+Book(int index, String title, String genre, double price, int author\_index, ArrayList<String> ExtraFields)

+toString() : String

+getIndex() : int

+getAuthorIndex : int

+getPrice() : double

+getTitle() : String

+getGenre() : String

+getAuthor : String

+setIndex(int index) : void

+setAuthorIndex(int authorIndex) : void

+setPrice(double price) : void

+setTitle(String title) : void

+setGenre(String genre) : void

+setAuthor(String Author) : void

**GUI**

+GUI(final String libraryString, final Library library)

-isInteger(String searchString) : boolean

*User Guide:*

Hello and welcome to the Library Information Management System. To begin, open the system and follow these instructions:

1. Select a .txt file with a list of books and authors to put in the system. The file must have the following format:

-Authors: a:index:name:address

-Books: b:index:title:genre:price:author index

The LIMS reads the file line by line. If any index is unknown, please enter “99999”. If any name, address, title or genre is unknown, please use “Unknown”. Extra items can be added to each line as long as a colon(:) separates each item. Spaces next to colons are ignored, but spaces inside titles, addresses, genres, and names are not. Any book that has an author index that doesn’t match any listed author’s index will be placed under the fake author “Unknown Author” who has an index of “99999”.

1. Once your file is uploaded and the LIMS initialized, you are ready to search through the books. A window will pop up with a search bar at the top. You can either search for a title, genre, or index number, or you can list the entire inventory by author with their corresponding books.
2. If you enter an index number or title, the LIMS will show one books with those credentials. If you search a genre, a list of books fitting that genre will appear.
3. If you click the Full List button, a list of authors with all their corresponding books will appear, starting with the unknown author continuing in the order the authors were uploaded from the file.

*Test Plan*

Input(.txt file):

b : 10001 : Java Basics : Science : 11.50 : 20003

b : 10002 : Advanced Java : Science : 10.99 : 20003

/b : 10003 : The Making of Apples : Science : 10.99 : 20001

b:10004:Mary Knows Best:Science:10.99:20002

a:20001:John Smith:Computers St. 50 Seattle

a : 20002 : Mary Jones : Literature Lane 25

Expected Full List Output:

Index: 999999999

Name: Unknown

Address: Unknown

Books:

Java Basics

Advanced Java

Number of Books: 2

Index: 20001

Name: John Smith

Address: Computers St. 50 Seattle

Books:

Number of Books: 0

Index: 20002

Name: Mary Jones

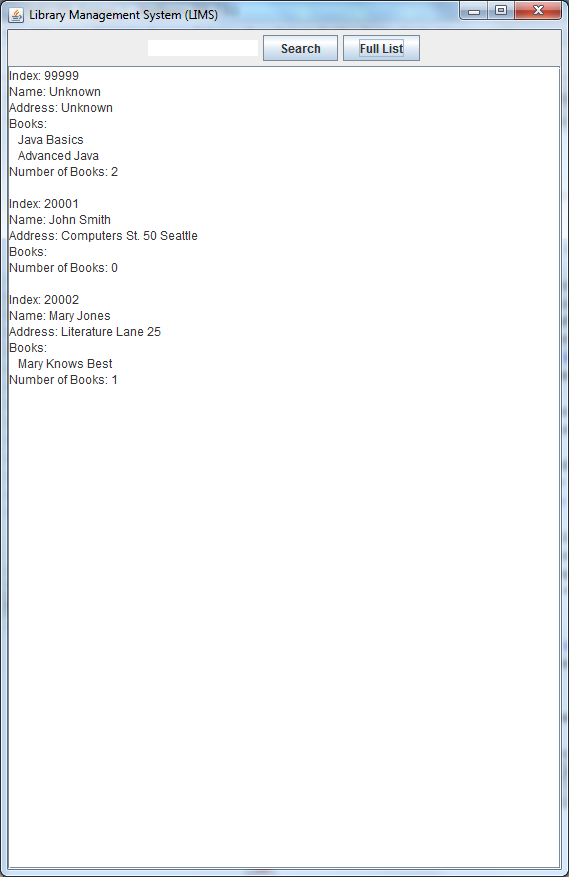
Address: Literature Lane 25

Books:

Mary Knows Best

Number of Books: 1

Actual Full List Output:



Search Bar Expected Output for book index 10001:

Index: 10001

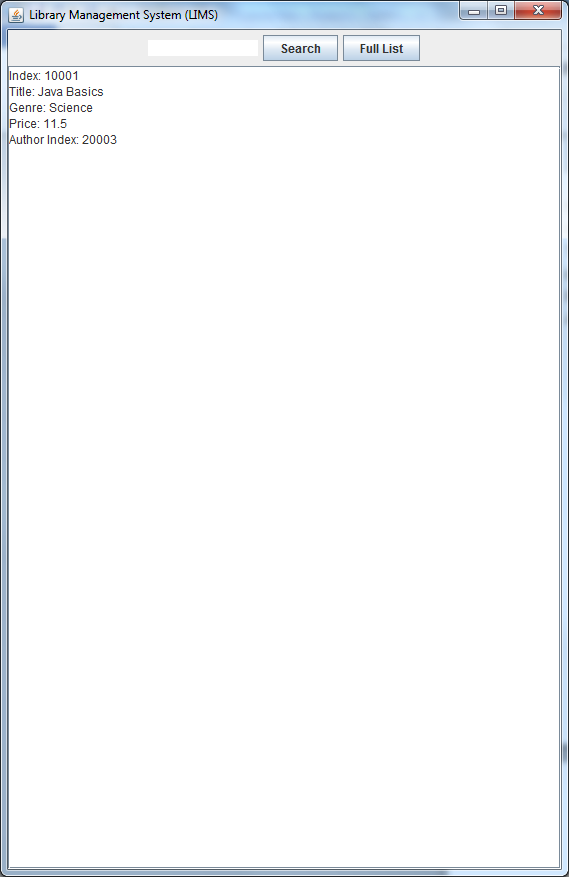
Title: Java Basics

Genre: Science

Price: 11.5

Author Index: 20003

Actual Output:



Expected Output Search Genre (Science):

Index: 10001

Title: Java Basics

Genre: Science

Price: 11.5

Author Index: 20003

Index: 10002

Title: Advanced Java

Genre: Science

Price: 10.99

Author Index: 20003

Index: 10004

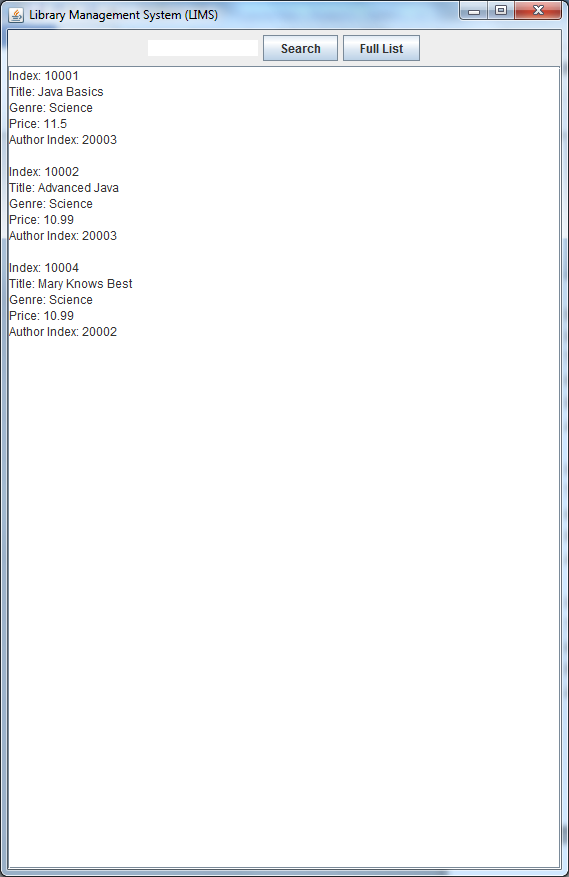
Title: Mary Knows Best

Genre: Science

Price: 10.99

Author Index: 20002

Actual Output:



*Comments:*

Plans for improvement in the future:

-More search parameters not just limited to title, genre, and index.

-More accessibility to the ExtraFields ArrayLists.

-More exception handling, the program is not currently very stable.

-More management than just books. Newspapers and magazines should be manageable.

-Many others, but these are the major ones.

*Lessons Learned:*

-The idea of an object holding a list of objects, each of which hold their own objects is something I needed practice on. I didn’t realize this actually constitutes a tree even though it’s not one of the formal trees such as BST or AVL.

-I’ve always struggled with the scope of variables, methods, classes. This project improved my ability to handle all of them.

*Areas to Improve:*

-While I believe I have a solid grasp of object oriented design, I still get nervous sometimes separating everything away from procedural design. My GUI class does not have a strong UML because I was nervous that declaring data fields outside of the constructor would end up with unusable code somehow. This is something I will need to work on.