Benjamin Knauth

CMSC 335

11 September 2014

Project #2 Description

*UMLs:*

**CMSC335\_Project\_1**

main(String args[]): void

**Library**

Authors : LinkedHashMap<Integer, Author>

+Library()

+toString() : String

+sortAuthorNames(LinkedHashMap<Integer, Author> authors) : LinkedHashMap<Integer, Author>

+sortAuthorIndexes(LinkedHashMap<Integer, Author> authors) : LinkedHashMap<Integer, Author>

**Author**

-index : int

-numberBooks : int

-name : String

-address : String

Books : LinkedHashMap<Integer, Book>

Journals : LinkedHashMap<Integer, Journal>

ExtraFields : ArrayList<String>

+Author()

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

+toString() : String

+sortBooksTitle(LinkedHashMap<Integer, Author> books) : LinkedHashMap<Integer, Author>

+sortBooksPrice(LinkedHashMap<Integer, Author> books) : LinkedHashMap<Integer, Author>

+sortBooksIndex(LinkedHashMap<Integer, Author> books) : LinkedHashMap<Integer, Author>

+sortJournalDate(LinkedHashMap<Integer, Author> journals) : LinkedHashMap<Integer, Author>

+sortJournalIssue(LinkedHashMap<Integer, Author> journals) : LinkedHashMap<Integer, Author>

+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

**Journal**

-issueNumber : int

-author\_index : int

-date : String

ExtraFields : ArrayList<String>

+Journal()

+Journal(String date, int issueNumber, int author\_index, ArrayList<String> ExtraFields)

+toString() : String

+getDate() : String

+getIssueNumber(): int

+getAuthorIndex() : int

+setDate(String newDate) : void

+setIssueNumber(int newIssueNumber) : void

+setAuthorIndex(int author\_index) : void

**GUI**

+GUI(final String libraryString, final Library library)

-isInteger(String searchString) : Boolean

**AuthorNameComparator**

+ compare(Author t1, Author t2) : int

**AuthorIndexComparator**

+ compare(Author t1, Author t2) : int

**BookTitleComparator**

+ compare(Book t1, Book t2) : int

**BookIndexComparator**

+ compare(Book t1, Book t2) : int

**BookPriceComparator**

+ compare(Book t1, Book t2) : int

**JournalDateComparator**

+compare(Journal t1, Journal t2) : int

**JournalIssueComparator**

+compare(Journal t1, Journal t2) : int

*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

-Journals: j:date:issue number:author index

(date needs to be in numerical form with a four digit year and two digit month, ie. 201405)

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 and Journals. 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 and journals will appear, arranged by author index.
4. The buttons on the left can be used to list authors, books, or journals by the listed method. For example, clicking All Authors – Name will produce a list of all authors in alphabetical order.

***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

a:33333:Amy Adams:Wisconsin Street

j:201401:5:20001

j:201405:222:20002

j:199404:99:20003

j:200012:40:20004

j:200110:1:33333

***Expected Full List Output:***

Index: 20001

Name: John Smith

Address: Computers St. 50 Seattle

Books:

Journals:

5

Number of Books: 0

Number of Journals: 1

Index: 20002

Name: Mary Jones

Address: Literature Lane 25

Books:

Mary Knows Best

Journals:

222

Number of Books: 1

Number of Journals: 1

Index: 33333

Name: Amy Adams

Address: Wisconsin Street

Books:

Journals:

1

Number of Books: 0

Number of Journals: 1

Index: 99999

Name: Unknown

Address: Unknown

Books:

Java Basics

Advanced Java

Journals:

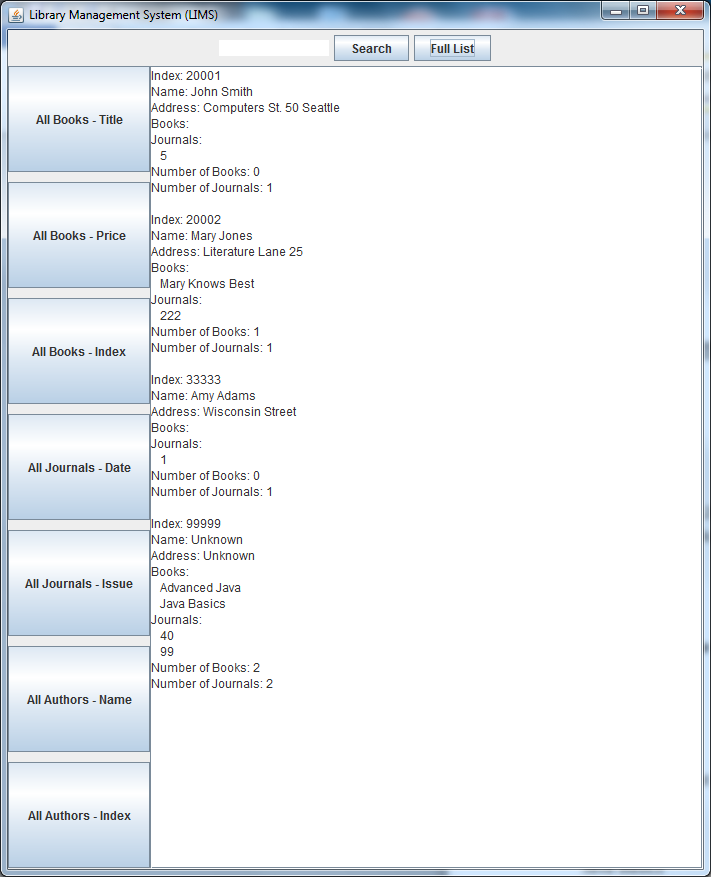
99

40

Number of Books: 2

Number of Journals: 2

***Actual Output***



***Expected All Books – Title***

Amy Adams:

John Smith:

Mary Jones:

Mary Knows Best

Unknown:

Advanced Java

Java Basics

***Actual Output***



***Expected All Books – Price***

Amy Adams:

John Smith:

Mary Jones:

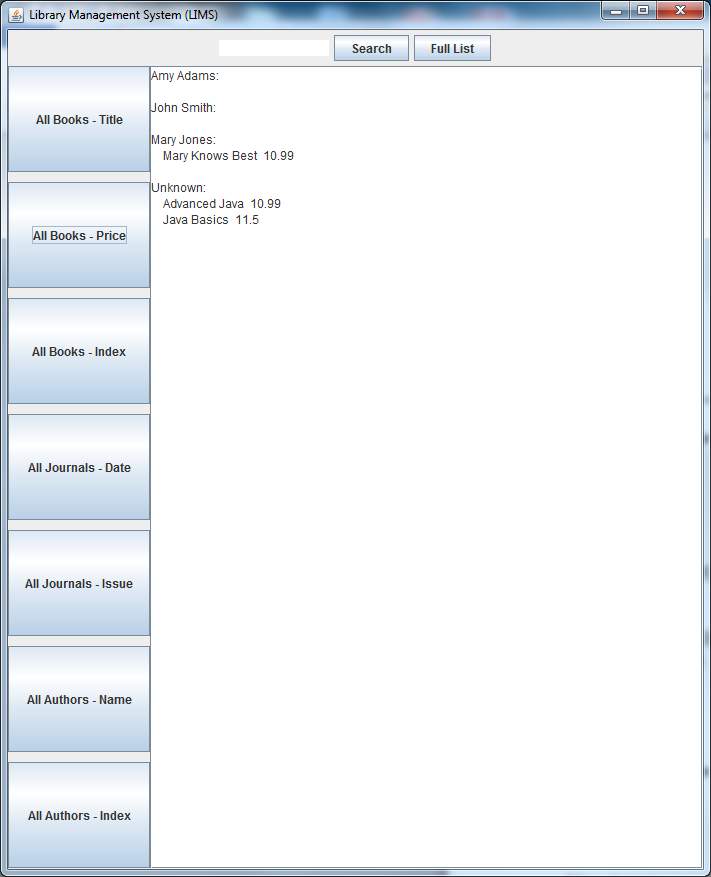
Mary Knows Best 10.99

Unknown:

Advanced Java 10.99

Java Basics 11.5

***Actual Output***



***Expected All Books – Index***

Amy Adams:

John Smith:

Mary Jones:

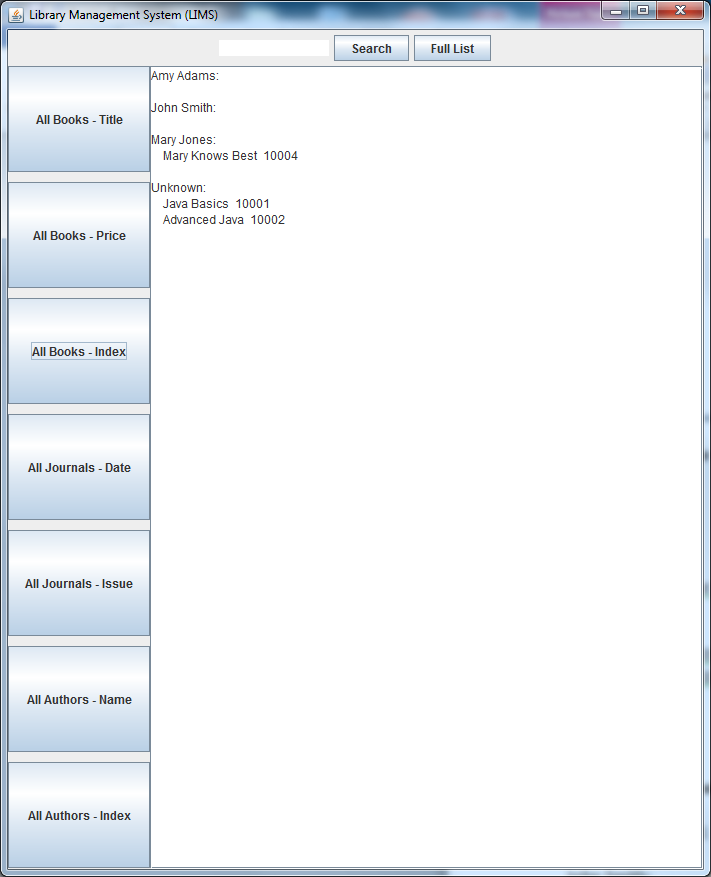
Mary Knows Best 10004

Unknown:

Java Basics 10001

Advanced Java 10002

***Actual Output***



***Expected All Journal - Date***

Amy Adams:

Issue Number: 1 Date: 200110

John Smith:

Issue Number: 5 Date: 201401

Mary Jones:

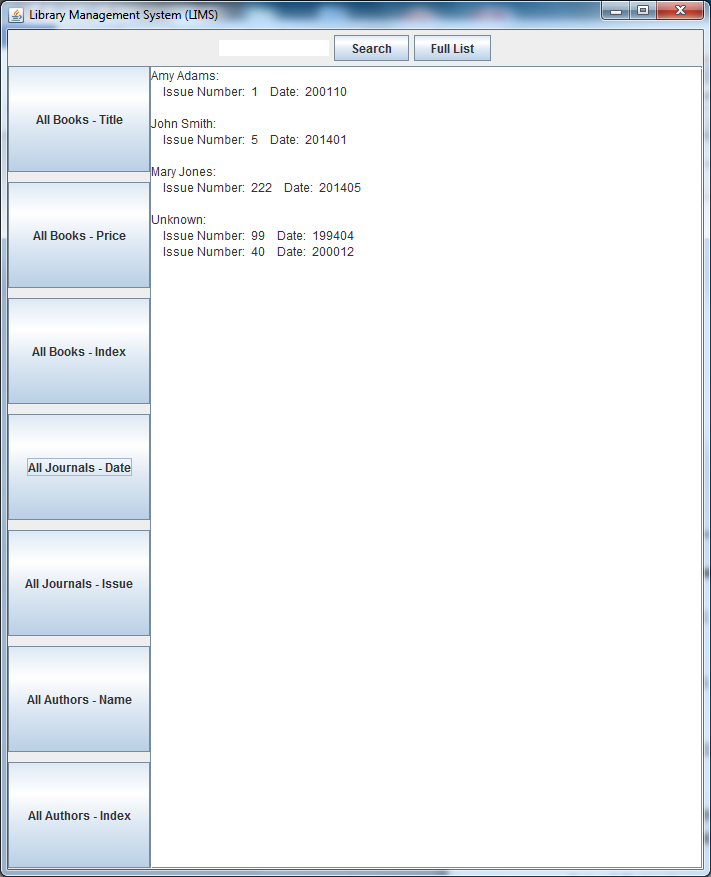
Issue Number: 222 Date: 201405

Unknown:

Issue Number: 99 Date: 199404

Issue Number: 40 Date: 200012

***Actual Output***



***Expected All Journal – Issue***

Amy Adams:

Issue Number: 1 Date: 200110

John Smith:

Issue Number: 5 Date: 201401

Mary Jones:

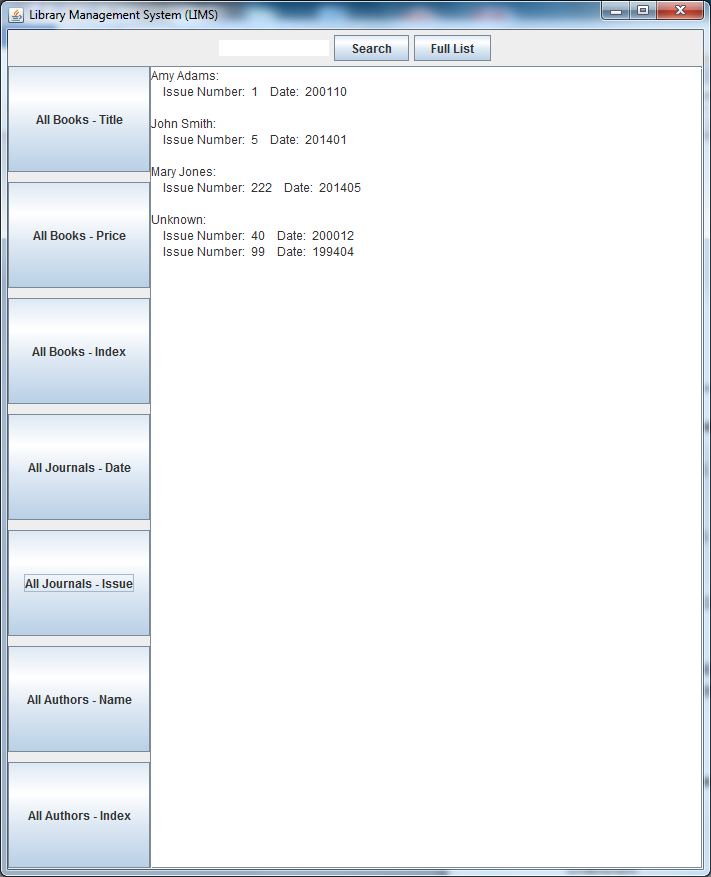
Issue Number: 222 Date: 201405

Unknown:

Issue Number: 40 Date: 200012

Issue Number: 99 Date: 199404

***Actual Output***



***Expected All Authors – Name***

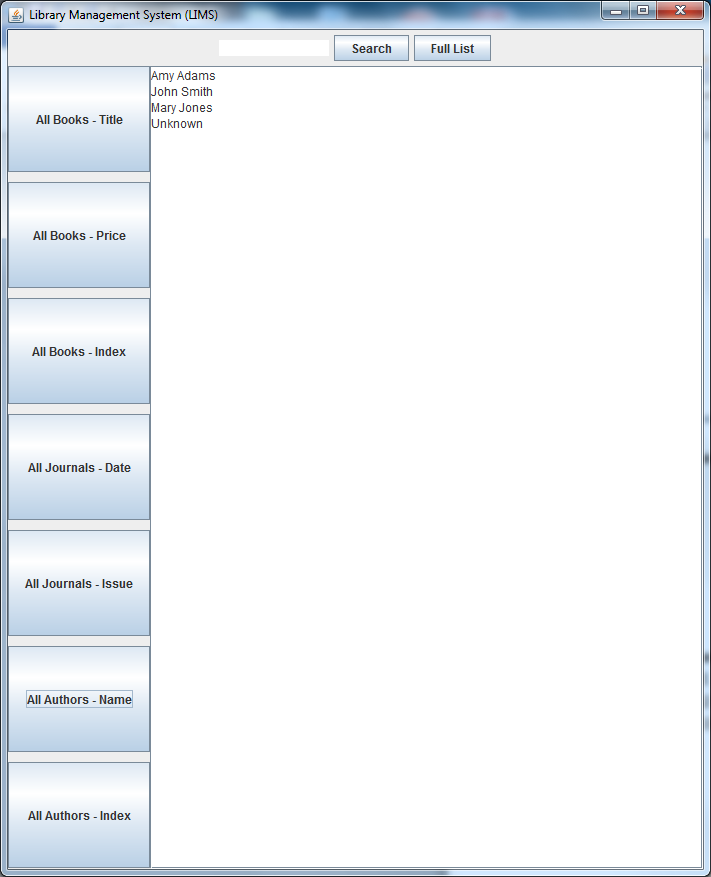
Amy Adams

John Smith

Mary Jones

Unknown

***Actual Output***



***Expected All Authors – Index***

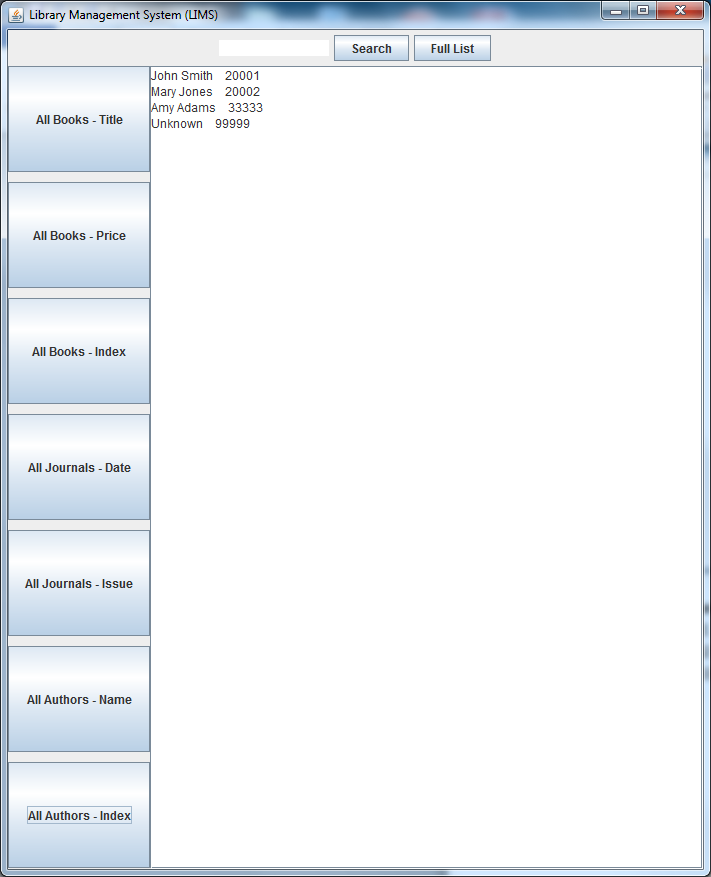
John Smith 20001

Mary Jones 20002

Amy Adams 33333

Unknown 99999

***Actual Output***



*Comments:*

Plans for improvement in the future:

-Expand Journal class to include Title

-Create Editor class to hold Journal objects

-More accessibility to the ExtraFields ArrayLists.

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

-Extend types of items beyond books and journals.

-Many others, but these are the major ones.

*Lessons Learned:*

-I’ve always struggled to sort objects based on object data fields, comparable has solved this.

-Getting lots of experience with this and Project 1 in using event listeners.

-Realized a mistake from last project; GUI(Library library, libraryString) was pointless, only needed GUI(Library library). I don’t know why I didn’t catch this.

*Areas to Improve:*

-Need more practice with various data structures.