Design  
Jack Jones

Contents

[Notes 4](#_Toc71535083)

[Resources 4](#_Toc71535084)

[Pseudocode 5](#_Toc71535085)

[Read File 5](#_Toc71535086)

[Write File 5](#_Toc71535087)

[Overwrite File 5](#_Toc71535088)

[Delete Record 6](#_Toc71535089)

[UML Class Diagrams 7](#_Toc71535090)

[Key 7](#_Toc71535091)

[Book 7](#_Toc71535092)

[BookRecordSystem 8](#_Toc71535093)

[FileHandling 8](#_Toc71535094)

[SceneBuilder 9](#_Toc71535095)

[StringManipulation 9](#_Toc71535096)

[Validation 9](#_Toc71535097)

[Flowcharts 10](#_Toc71535098)

[validISBN 10](#_Toc71535099)

[reverseString 11](#_Toc71535100)

[Form Design 12](#_Toc71535101)

[Home 13](#_Toc71535102)

[Add Record 13](#_Toc71535103)

[View All Records 14](#_Toc71535104)

[View Specific Record 14](#_Toc71535105)

[Select Item To Edit 15](#_Toc71535106)

[Edit Record 15](#_Toc71535107)

[Delete Record 16](#_Toc71535108)

[Search Form 17](#_Toc71535109)

[Search Results 17](#_Toc71535110)

[**Stats Home** 18](#_Toc71535111)

[Pie Chart 18](#_Toc71535112)

[Line Graph 19](#_Toc71535113)

[Averages 19](#_Toc71535114)

[Code Changes 20](#_Toc71535115)

[Search File Pseudocode 20](#_Toc71535116)

[Original 20](#_Toc71535117)

[New 21](#_Toc71535118)

# Notes

* Items in red text or surrounded by a red rectangle are added after the project has started.

# Resources

* <https://docs.oracle.com/javafx/2/get_started/hello_world.htm>
  + JavaFX Set up
* <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/TableView.html>
  + How to use a TableView in JavaFX
* <https://www.instructables.com/How-to-verify-a-ISBN/>
  + How to verify an ISBN
* <https://docs.oracle.com/javafx/2/charts/pie-chart.htm>
  + JavaFX Pie Chart
* <https://docs.oracle.com/javafx/2/charts/line-chart.htm>
  + JavaFX Line Graph
* <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Alert.html>
  + Used for errorBox method
* <https://stackoverflow.com/a/15801999/13644957>
  + String is Integer regex

# Pseudocode

## Read File

function readFile(){

    Scanner scan = new Scanner(filename);

    toWrite = "";

    while(scan.hasNextLine()){

        toWrite += scan.nextLine()

    }

    return toWrite

}

## Write File

function writeFile(content){

    newContent = readFile();

    toWrite = newContent + "\n" + content;

    writeFile(toWrite);

}

## Overwrite File

function overwriteFile(content){

    file.write(content)

}

## Delete Record

function deleteRecord(id){

    originalContents = readFile();

    toWrite = "";

    for line in originalContents{

        if(id not in line){

            toWrite += line;

        }

    }

    overwriteFile(toWrite);

}

# UML Class Diagrams

## Key

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| + | Public |
| - | Private |
| # | Protected |

## Book

|  |
| --- |
| **Book** |
| -id: String  -isbn: String  -title: String  -author: String  -pages: String  -genre: String  -review: String |
| +Book(): Constructor  +getId(): String  +setId(String): Void  +getIsbn(): String  +setIsbn(String): Void  +getTitle(): String  +setTitle(String): Void  +getAuthor(): String  +setAuthor(String): Void  +getPages(): String  +setPages(String): Void  +getGenre(): String  +setGenre(String): Void  +getReview(): String  +setReview(String): Void  +toString(): String |

## BookRecordSystem

|  |
| --- |
| **BookRecordSystem** |
|  |
| +main(String[]): Void  +start(Stage): Void |

## FileHandling

|  |
| --- |
| **FileHandling** |
| -filename: String  -file: File  -stringManipulator: stringManipulation  -validator: Validation |
| +FileHandling(String):Constructor  +getFilename(): String  +setFilename(String): Void  +writeLineToFile(String): Void  +overwriteFile(String): Void  +readAllFile(): String  +deleteRecord(String): Void  +editRecord(String, String): Void  +isDuplicate(String, Boolean): Boolean  +getNextId(): String  +searchFile(String[]): ArrayList<String>  +countOccurences(int): Map<String, Integer>  + getTwoIntFields(int, int): Map<Integer, Integer>  +getStringMode(int): String  +getDoubleMean(int): double  +getDoubleMedian(int): double  +getIntRange(int): int |

## SceneBuilder

|  |
| --- |
| **SceneBuilder** |
| -validator: Validation  -fileHandler: FileHandling  -primaryStage: Stage  -sceneHome: Sceme |
| +SceneBuilder(Stage): Constructor  + getSceneHome(): Scene  - getSceneAdd(): Scene  - getSceneViewAll(): Scene  - getSceneViewSpecific(Book): Scene  - getSceneSelectEdit(): Scene  - getSceneEditSpecific(Book): Scene  - getSceneDelete(): Scene  - getSceneSearchForm(): Scene  - getSceneSearchResult(String[]): Scene  - getSceneStatsHome(): Scene  - getSceneGenreStat(): Scene  - getScenePageStat(): Scene  - getSceneAverages(): Scene  - errorBox(String, String): Void  -getTable(String[]):TableView<Book>  - getObservableList(String[]):ObservableList<Book> |

## StringManipulation

|  |
| --- |
| **StringManipulation** |
|  |
| + StringManipulation(): Constructor  +csvColumnEquals(String, int, String): Boolean  + reverseString(String): String |

## Validation

|  |
| --- |
| **Validation** |
| - stringManipulator: StringManipulation |
| + stringIsInteger(String): boolean  + validIsbn(String): boolean  + validIsbnTenDigit (String): boolean  + validIsbnThirteenDigit (String): boolean  + validStringLength (String, int, int): boolean  + validTitle (String): boolean  + validAuthor (String): boolean  + validPages (String): boolean |

# Flowcharts

## validISBN

Diagram

Description automatically generated

## reverseString

Diagram

Description automatically generated

# Form Design

**NOTE: See Also form-walkthrough.pptx**

**Double click below**



## Home

Graphical user interface

Description automatically generated

## Add Record

Graphical user interface

Description automatically generated

## View All Records

Graphical user interface

Description automatically generated with medium confidence

## View Specific Record

Graphical user interface

Description automatically generated

## Select Item To Edit

Graphical user interface

Description automatically generated with medium confidence

## Edit Record

Graphical user interface

Description automatically generated

## Delete Record

Graphical user interface

Description automatically generated

## Search Form

Graphical user interface

Description automatically generated

## Search Results

Graphical user interface

Description automatically generated

## **Stats Home**

A picture containing diagram

Description automatically generated

## Pie Chart

A picture containing graphical user interface

Description automatically generated

## Line Graph

A picture containing graphical user interface

Description automatically generated

## Averages

Graphical user interface

Description automatically generated

# Code Changes

## Search File Pseudocode

### Original

function searchFile(String[] details){

    String[] toSearch = readFile().split("\n");

    ArrayList<String> toReturn = new ArrayList<>();

    for line in toSearch:

        String[] lineArray = toSearch.split(",");

        if (lineArray[0] == details[0] and details[0].length() > 0) or (lineArray[1] == details[1] and details[1].length() > 0) or

        // Futher Code

        or (lineArray[6] == details[6] and details[6].length() > 0):

            toReturn.add(line)

    return toReturn

}

This code didn’t work. It did not allow for multifield searching, and so as a result incorrect results were being returned.

### New

function searchFile(String[] details){

    int needed = 0

    for item in details:

        if item.length() > 0:

            needed += 1

    String[] toSearch = readFile().split("\n")

    ArrayList<String> toReturn = new ArrayList<String>()

if needed == 0:

return toReturn

    int matched = 0

    for line in toSearch:

        matched = 0

        String[] lineArray = line.split(",")

        if(details[0].toLowerCase() in lineArray[0].toLowercase() and details[0].length() > 0):

            matched += 1

        // More Code

        if(details[6].toLowerCase() in lineArray[6].toLowercase() and details[6].length() > 0):

            matched += 1

        if matched == needed:

            toReturn.add(line)

    return toReturn

}

* To work out how many fields have been searched, the function counts how many items in the details array has a length greater than 0.
* The function then loops through the file contents and takes each line as an array.
* If the item in the same position of the details array is equal to that of the line array, the matched variable in incremented by 1.
* If the matched variable is equal to the amount needed, the line is added to the toReturn arraylist