### **60%**

### **Hands-On Test: Bookstore Inventory Search Application**

**Submission Requirements:**

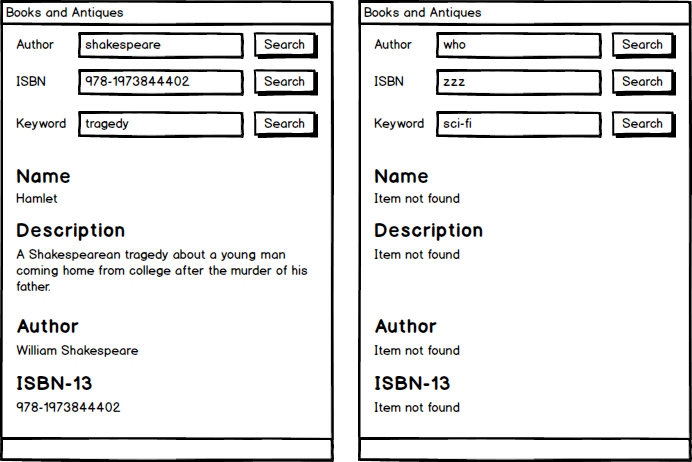
* Place your completed solution in your **AllCoursework Repository** under the HandsOnTest/CH06/ folder.

**General Guidelines:**

* You may use your textbook, lab materials, and the internet as resources.
* All code submitted must be your original work.

### **[PART A - 100 pts] GUI Application: Bookstore Inventory Search**

Develop a Graphical User Interface (GUI) application that allows a user to search a bookstore's inventory of books.



**Core Functionality:** The application must support the following search capabilities:

* **Search by Author Name:** Users can search for books by author name (case-insensitive, allows partial matches).
* **Search by ISBN Number:** Users can search for books by ISBN-13 number.
* **Keyword Search:** Users can perform a keyword search that checks both the Name and Keywords fields of a book (case-insensitive, allows partial matches).

**Data Storage:**

* The program must store the book inventory data using **parallel arrays or Lists**.
* Utilize appropriate loops for data processing and searching.

**Book Inventory Data:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Keywords** | **Author** | **ISBN-13** |
| The Great Gatsby | The story of eccentric millionaire Jay Gatsby and his pursuit of his lost love. | F. Scott Fitzgerald | 978-1847496140 |
| War and Peace | A fictional story about the 1812 French invasion of Russia. | Leo Tolstoy | 978-1400079988 |
| Moby-Dick | The story of a sailor’s relentless hunt for a white whale. | Herman Melville | 978-1503280786 |
| Hamlet | A Shakespearean tragedy about a young man coming home from college after the murder of his father. | William Shakespeare | 978-1973844402 |
| Pride and Prejudice | A comedic story of love and life in Old England. | Jane Austen | 978-0141439518 |

**Required Methods:**

Implement the following methods as described:

1. **ShowBook(index)**
   * **Accepts:** An integer index.
   * **Functionality:** Displays all the information (Name, Keywords, Author, ISBN-13) about the book at the given index.
   * **Error Handling:** If the index is invalid (e.g., out of bounds), display appropriate error messages.
2. **SearchByAuthor(authorName)**
   * **Accepts:** A string authorName.
   * **Returns:** The integer index of the *first* matching book found.
   * **Functionality:** Searches the inventory for books by the provided author name. Must be case-insensitive and allow partial matches.
3. **SearchByIsbn(isbnNumber)**
   * **Accepts:** A string isbnNumber.
   * **Returns:** The integer index of the *first* matching book found.
   * **Functionality:** Searches the inventory for books by the provided ISBN-13 number.
4. **SearchByKeyword(keyword)**
   * **Accepts:** A string keyword.
   * **Returns:** The integer index of the *first* matching book found.
   * **Functionality:** Searches the Name and Keywords fields for the provided keyword. Must be case-insensitive and allow partial matches.

### **Grading Rubric (100 pts) -** Each bullet item must fully meet requirements to receive points, no partial points.

**Code Structure & Conventions (15 pts)**

* **5 pts:** Control names follow established naming conventions.
* **5 pts:** Variable names follow established naming conventions.
  + string[] KeyWords = new string[]
* **5 pts:** Method names follow established naming conventions.

**Data Management (5 pts)**

* **5 pts:** Program stores book information in parallel arrays or Lists as required.

**Search Functionality (45 pts)**

* **10 pts:** User can successfully search for a book by author name.
* **10 pts:** User can successfully search for a book by ISBN number.
* **10 pts:** User can successfully search for a book by keyword.
* **5 pts:** Author name search is case-insensitive and allows partial matches.
* **5 pts:** Keyword search is case-insensitive and allows partial matches.
* **5 pts:** Keyword search correctly looks at both the Name and Keywords fields of the book.

**Output & Error Handling (15 pts)**

* **5 pts:** Successful searches consistently display all information about the found book (via ShowBook method).
* **10 pts:** Failed searches consistently display all expected error messages.

**Method Implementation (20 pts)**

* **4 pts:** ShowBook() method is implemented as described. – Not used
* **4 pts:** SearchByAuthor() method is implemented as described. Did not properly use loop as described
* **4 pts:** SearchByIsbn() method is implemented as described. Did not properly use loop as described
* **4 pts:** SearchByKeyword() method is implemented as described. Did not properly use loop as described
* **4 pts:** Return values from each custom search method (SearchByAuthor, SearchByIsbn, SearchByKeyword) are correctly utilized in the calling method to display results or error messages.