Tuesday, September 5, 2023

Library Management System Software Development Plan

Eric Quispe

Software Development I

CRN:16046

1. Introduction

This document outlines the software development plan for creating a Library Management System (LMS). The purpose of this software is to help manage the collection of books at a library.

2. Define Requirements (updates to features added as the class continues)

The LMS will a console menu with the following features:

1. Add new books to the collection from a text file.
2. Remove a book from the collection using its ID number.
3. List all books currently in the collection.

System Users and Tasks

The intended primary users of the LMS will be library staff and administrators. They will interact with the system to:

- Add new books to the collection.
- Remove books from the collection.
- View the list of all books in the collection.

Constraints

The software must operate under the following constraints:

- The input text file must be formatted with book details separated by commas.
- Each book has a unique ID, title, and author.

3. Gather Requirements

Intended users will be library staff and administrators which will give feedback on the interaction and provide us with information on what other functions they would need in the software. The interaction with the LMS involves:

- Providing a text file with book details to be added to the collection.
- Specifying the ID number of a book to be removed.
- Requesting to view the list of all books in the collection.

4. Implementation Plan

Task Breakdown

1. Create a console menu
2. Read input text file and extract book details.
3. Create a Book class with attributes: ID, title, and author.
4. Implement a method to add a new book to the collection.
5. Implement a method to remove a book from the collection using its ID.
6. Implement a method to list all books in the collection.
7. Assemble the main program that interacts with the user and performs actions.

Class and Method Details

- **LibraryManagementSystem Class**
  - Methods:
    - `getUsersChoice`(): allow the user to pick from the console menu
    - `addBook`(): Adds a new book to the collection based on the text file. This also contains the bufferReader to get information from the text file.
    - `removeBook(int bookId)`: Removes a book from the collection based on its ID.
    - `listAllBooks()`: Lists all books in the collection.

- **Book Class**
  - Attributes: ID (int), title (String), author (String)
  - Methods:
    - `getters`()
- **Library Class**
  - Methods:
    - `addBook(Book book)`: Adds a new book to the collection based on the text file.
    - `removeBook(int bookId)`: Removes a book from the collection based on its ID.
    - `listAllBooks()`: Lists all books in the collection.

`addBook` Method

1. Receive a `Book` object as an argument.
2. Append the `Book` object to the collection.

`removeBook` Method

1. Receive a book ID as an argument.

2. Search the collection for the book with the given ID.
3. If found, remove the book from the collection.

`listAllBooks` Method

1. Loop through the collection of books.
2. Print each book's details (ID, title, author).

5. Testing Plan

Test Cases

1. Test adding a new book to the collection.
2. Test removing a book from the collection.
3. Test listing all books in the collection.

6. Deploy Software

Please refer to the attached code for the implementation of the LMS. The code has been structured according to the implementation plan.

Class LMS

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.*;

/**
 * Eric Quispe
 * Course:202410 Software Development I CEN-3024C-16046
 * Date: 9/6/23
 *
 * LMS class runs the Library Management System program were the user will
have the most interaction with.
 *  It reads book data from a text file books.txt, allows adding and removing
books,
 *  and lists all books in the library's collection.
 */

public class LibraryManagementSystem {

      public static void main(String[] args)
      {
        Library library = new Library();
        boolean exit = false;
```

```
        while (!exit) {
            System.out.println("\nLibrary Management System Menu:");
            System.out.println("1. Add a Book");
            System.out.println("2. Remove a Book");
            System.out.println("3. List All Books");
            System.out.println("4. Exit");

            int choice = getUserChoice();

            switch (choice) {
                case 1:
                    addBook(library);
                    break;
                case 2:
                    removeBook(library);
                    break;
                case 3:
                    listAllBooks(library);
                    break;
                case 4:
                    exit = true;
                    System.out.println("Exiting the Library Management
System. Goodbye!");
                    break;
                default:
                    System.out.println("Invalid choice. Please select a valid
option.");
            }
        }

    }

      // getUserChoice() will allow the user to enter a choice from the menu
      private static int getUserChoice()
      {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your choice: ");
        while (!scanner.hasNextInt())
        {
            scanner.next(); // Clear the invalid input
            System.out.print("Invalid input. Please enter a valid number: ");
        }
        return scanner.nextInt();
    }

      // addBook method will add a book based off the book.txt file
      private static void addBook(Library library)
      {
            // Read book data from text file and add to the collection
        try (BufferedReader reader = new BufferedReader(new
FileReader("C:\\Users\\uncha\\OneDrive\\Desktop\\books.txt"))) {
            String line;

            while ((line = reader.readLine()) != null)
            {
```

```
                String[] bookData = line.split(",");// breaking the strings
by looking for "," in the text file
                int id = Integer.parseInt(bookData[0]); // assigning Id
                String title = bookData[1]; // assigning title
                String author = bookData[2]; // assigning author
                Book book = new Book(id, title, author); // creating a new
book
                library.addBook(book);// adding to the colleciton
            }

        }
        catch (IOException e) // in the event file was not read
        {
            System.out.println("Error reading file: " + e.getMessage());
        }
      System.out.print("\n Books have been added to the collection via the
book.txt file");
      }

      // This method will allow the user to enter a book to remove from the
library
       private static void removeBook(Library library)
       {
              Scanner scanner = new Scanner(System.in);
              System.out.print("Enter book ID to remove: ");
              int bookId = scanner.nextInt();
              library.removeBook(bookId);// this will refer to the library
method to remove the book
              System.out.println("\nBook removed from the collection.");
        }
       // Display all books in the collection
       private static void listAllBooks(Library library)
         {
              System.out.println("\nAll books in the collection:");
              library.listAllBooks();// refers to library method to show all
books in the collection
          }



}// end of LMS class
```

Class library

```
import java.util.ArrayList;
import java.util.List;
public class Library
/**
 * Eric Quispe
 * Course:202410 Software Development I CEN-3024C-16046
 * Date: 9/6/23
 *
 * Library class will manage the collection of books
 */
```

```
{
       private List<Book> bookCollection = new ArrayList<>();// this list
hold Books

          // Method to add a book to the collection
          public void addBook(Book book)
          {
              bookCollection.add(book);
          }

          /* Method to remove a book from the collection by ID which is
assigned in the LMS class
          *  removeBook method also has a feature to let the user know that a
book wasn't found based on the ID
          */
          public void removeBook(int bookId)
          {
              Book foundBook = null;
              for (Book book : bookCollection) {
                  if (book.getId() == bookId) {
                      foundBook = book;
                      break;
                  }
            }
              if (foundBook != null) {
                  bookCollection.remove(foundBook);
              } else {
                  System.out.println("Book with ID " + bookId + " not
found.");
              }
          }

          // Method to list all books in the collection
          public void listAllBooks() {
              for (Book book : bookCollection) {
                  System.out.println("ID: " + book.getId() + ". Title: " +
book.getTitle() + ", Author: " + book.getAuthor());
              }
          }

}
```

Class book

```
/**
 * Eric Quispe
 * Course:202410 Software Development I CEN-3024C-16046
 * Date: 9/6/23
 *
```

```java
 * Book class is used to create the books based on what is enter on the text
file.
 * here is where the book is assigned id, Title, and author
 */

public class Book
{
      private int id;
    private String title;
    private String author;

    public Book(int id, String title, String author) {
        this.id = id;
        this.title = title;
        this.author = author;
    }

    // These methods just get the information stated ID,Title, or Author
    public int getId()
    {
        return id;
    }

    public String getTitle()
    {
        return title;
    }

    public String getAuthor()
    {
        return author;
    }
}
```

Output so far

```
Library Management System Menu:
1. Add a Book
2. Remove a Book
3. List All Books
4. Exit
Enter your choice: 1

 Books have been added to the collection via the book.txt file
Library Management System Menu:
1. Add a Book
2. Remove a Book
3. List All Books
4. Exit
Enter your choice: 3

All books in the collection:
ID: 1. Title: To Kill a Mockingbird, Author: Harper Lee
ID: 2. Title: 1984, Author: George Orwell
```

```
ID: 3. Title: The Great Gatsby, Author: F. Scott Fitzgerald

Library Management System Menu:
1. Add a Book
2. Remove a Book
3. List All Books
4. Exit
Enter your choice: 2
Enter book ID to remove: 1

Book removed from the collection.

Library Management System Menu:
1. Add a Book
2. Remove a Book
3. List All Books
4. Exit
Enter your choice: 3

All books in the collection:
ID: 2. Title: 1984, Author: George Orwell
ID: 3. Title: The Great Gatsby, Author: F. Scott Fitzgerald

Library Management System Menu:
1. Add a Book
2. Remove a Book
3. List All Books
4. Exit
Enter your choice:
```