Quintin Johnson

The LMS Project

Software Development I

CRN 3024C

9/9/2024

Planning

This LMS will have a user login and a simple GUI that the user can use to search for books. The system will give each book a unique id number to locate the book. The system will also allow people to add more books to the database. I will use Valencia’s LMS, Cengage, Udemy and my local library’s LMS to give me an idea of how to build this. The goal of this project is to make it easier for people to search for a book instead of going to a bookshelf and looking for it.

Defining Requirements

1. Each book in the system has a unique ID number. Each book also has a title and author.
2. Users must be able to add new books to the collection from a text file.
3. Users must be able to remove a book from the collection using its ID number.
4. Users must be able to see a list of all books currently in the collection.
5. Users must interact with the system using an on-screen menu.

Designing

I will have one page for the LMS. You will be able to search for a book with the title or by book id. You can check out or remove a book from the system by looking it up with the title or book id. You can add a book a to the system. Here is a rough idea of how the system will work and look.

LMS Home Page

Quit

Remove Book By Title

Remove Book By Id

Search by Book Id

Search by title

Add New Book

Show All Books

When User Clicks on Search by Title

Enter Title: 48 Laws of Power

Title: The Bible

ID: 01

Author: King James Version

Copies: 7

Adding a Book

Book ID: 48 laws of powers by Robert Greene

Clear

OK

Thank You! You have successfully added the

48 laws of powers by Robert Greene.

Building

I will use Java to create the Gui and I will use a Hash-map to store the books in a database. The Gui for the LMS home page will have the text fields which will be book id, title and author. I will have seven buttons in the main menu. The first button will give the user the option to see all the books we have in stock. The second button will search for the book by the title. The third button will search the book with its book id. The fourth button is an add book button. The fifth button will remove the book with the book id. The sixth button will remove the book by searching the title. My seventh button will let the user quit the program.

Testing

I will test the system every 3-5 days before the due date to ensure that the LMS is working properly. I will take down any notes that are causing issues with my code. I will also have 1-2 other testers try out the LMS to get feedback on it.

Maintenance

I will maintain this LMS to see if there are any additional features that I can add. I would like to add small details with my buttons. I would like to have the button look like it has been pushed when it is clicked. Second, I would like to add a hover effect or a shadow when the user’s mouse is on the button.

Deployment

/\*  
Quintin Johnson  
Sep 9,2024  
Cen 3024  
  
This class contains the Gui methods that  
give you the option to search for a book by title,  
or book id. You can also remove books and see all the books that  
we have in stock.  
  
 \*/  
import java.awt.\*;  
import javax.swing.\*;  
import java.util.HashMap;  
import java.util.Map;  
  
  
public class Gui extends JFrame {  
 public static void main(String[] args) {  
 // Calling The GUI  
 new Gui();  
  
  
 }  
  
 //Setting up the objects for the Gui  
 private JFrame frame;  
 private JPanel mainMenuPanel;  
 private JTextArea textArea;  
 private JTextField inputField;  
 private HashMap<String, bookDetails> books = new HashMap<>();  
 private HashMap<String, bookDetails> books2 = new HashMap<>();  
  
 public Gui() {  
 // Calling Book DataBase  
 Books();  
  
 // Creating the frame  
 frame = new JFrame("LMS");  
 frame.setSize(400, 400);  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.setLayout(new BorderLayout());  
  
 // Text area to display  
 textArea = new JTextArea();  
 textArea.setEditable(false);  
 JScrollPane scrollPane = new JScrollPane(textArea);  
  
 // Input user text  
 inputField = new JTextField();  
 inputField.setPreferredSize(new Dimension(200, 25));  
  
 // The main menu layout  
 mainMenuPanel = new JPanel();  
 mainMenuPanel.setLayout(new GridLayout(8, 1));  
  
 // Buttons for main menu  
 JButton showAllBooksBtn = new JButton("Show All Books");  
 JButton searchTitleBtn = new JButton("Search Book By Title");  
 JButton searchIdBtn = new JButton("Search Book By ID");  
 JButton addBookBtn = new JButton("Add a New Book");  
 JButton removeByIdBtn = new JButton("Remove Book By ID");  
 JButton removeByTitleBtn = new JButton("Remove Book By Title");  
 JButton quitBtn = new JButton("Quit");  
  
 // Adding buttons to panel  
 mainMenuPanel.add(showAllBooksBtn);  
 mainMenuPanel.add(searchTitleBtn);  
 mainMenuPanel.add(searchIdBtn);  
 mainMenuPanel.add(addBookBtn);  
 mainMenuPanel.add(removeByIdBtn);  
 mainMenuPanel.add(removeByTitleBtn);  
 mainMenuPanel.add(quitBtn);  
  
 // Action listeners for buttons  
 showAllBooksBtn.addActionListener(e -> showAllBooks());  
 searchTitleBtn.addActionListener(e -> searchByTitle());  
 searchIdBtn.addActionListener(e -> searchByID());  
 addBookBtn.addActionListener(e -> addBook());  
 removeByIdBtn.addActionListener(e -> removeBookByID());  
 removeByTitleBtn.addActionListener(e -> removeBookByTitle());  
 quitBtn.addActionListener(e -> System.*exit*(0));  
  
 // Adding panels to frame  
 frame.add(mainMenuPanel, BorderLayout.*WEST*);  
 frame.add(scrollPane, BorderLayout.*CENTER*);  
 frame.add(inputField, BorderLayout.*SOUTH*);  
  
 //   
 frame.setVisible(true);  
 }  
  
 // My Book Database  
 private void Books() {  
 books.put("The Bible", new bookDetails("The Bible", "01", "King James Version", 7));  
 books.put("48 Law Of Power", new bookDetails("48 Law Of Power", "02", "Robert Greene", 2));  
 books.put("Curtis", new bookDetails("Curtis", "03", "50 Cent", 2));  
 books.put("The Godfather", new bookDetails("The Godfather", "04", "Mario Puzo", 2));  
 books.put("My New Story", new bookDetails("My New Story", "05", "Smooth Beats", 4));  
 books.put("Art of Seduction", new bookDetails("Art of Seduction", "06", "Robert Greene", 3));  
 books.put("Pride and Prejudice", new bookDetails("Pride and Prejudice", "07", "Jane Austen", 2));  
 books.put("The Adventures of Huckleberry Finn", new bookDetails("The Adventures of Huckleberry Finn", "08", "Mark Twain", 3));  
 books.put("Blue Ocean Strategy", new bookDetails("Blue Ocean Strategy", "09", "Renée Mauborgne", 2));  
 books.put("The Power of Discipline", new bookDetails("The Power of Discipline", "10", "Daniel Walter", 3));  
  
 // Adding books to database  
 for (Map.Entry<String, bookDetails> entry : books.entrySet()) {  
 books2.put(entry.getValue().getbookId(), entry.getValue());  
 }  
 }  
  
 // Show all books in database  
 private void showAllBooks() {  
 textArea.setText(" ");  
 //Looks For books and displays them  
 for (bookDetails book : books.values()) {  
 textArea.append(book.toString() + "\n");  
 }  
 }  
  
 // This searches a books by its title by asking the user  
 // to input the book its looking for  
 private void searchByTitle() {  
 String title = JOptionPane.*showInputDialog*("Enter Title:");  
 bookDetails book = books.get(title);  
 // If book is not null it will search and pull up the title  
 //If it is null the else runs  
 if (book != null) {  
 textArea.setText("Title: " + book.gettitle() + "\n" +  
 "ID: " + book.getbookId() + "\n" +  
 "Author: " + book.getauthor() + "\n" +  
 "Copies: " + book.getcopiesInstock());  
 } else {  
 textArea.setText("Your book is not here.");  
 }  
 }  
  
 // Search books by ID and prompts the user to enter a  
 // book id number  
 private void searchByID() {  
 String bookId = JOptionPane.*showInputDialog*("Enter Book ID:");  
  
 // User puts in data & it searches for the book in the Hashmap  
 // If we find a book it displays it. If not then the else runs  
  
 bookDetails book = books2.get(bookId);  
 if (book != null) {  
 textArea.setText("Title: " + book.gettitle() + "\n" +  
 "ID: " + book.getbookId() + "\n" +  
 "Author: " + book.getauthor() + "\n" +  
 "Copies: " + book.getcopiesInstock());  
 } else {  
 textArea.setText("Your book is not here.");  
 }  
 }  
  
  
 // The system ask for the Title, Book id, Author & Copies  
 // for the user to input  
 private void addBook() {  
 String bookId = JOptionPane.*showInputDialog*("Enter Book ID:");  
 String title = JOptionPane.*showInputDialog*("Enter Title:");  
 String author = JOptionPane.*showInputDialog*("Enter Author:");  
 int copies = Integer.*parseInt*(JOptionPane.*showInputDialog*("Enter Copies:"));  
  
 // This code Adds the new book in the Hashmap  
 books.put(title, new bookDetails(title, bookId, author, copies));  
 books2.put(bookId, new bookDetails(title, bookId, author, copies));  
 textArea.setText("Book Added: " + title);  
 }  
  
 // Asks user for input to remove book  
 private void removeBookByID() {  
 String bookId = JOptionPane.*showInputDialog*("Enter Book ID to Remove:");  
 bookDetails removedBook = books2.remove(bookId);  
  
 // when user adds a book it searches for the book in the Hashmap  
 // & it removes the book by its ID  
 if (removedBook != null) {  
 books.remove(removedBook.gettitle());  
 textArea.setText("Book Removed: " + removedBook.gettitle());  
 } else {  
 textArea.setText("Your book is not here.");  
 }  
 }  
  
 // This Searches the book in the Hashmap by the title to start the removal process  
 private void removeBookByTitle() {  
 String title = JOptionPane.*showInputDialog*("Enter Book Title to Remove:");  
 bookDetails removedBook = books.remove(title);  
  
 // After removedBook is found in the hashamp it is then deleted  
 // If it cant be found then the else will run  
 if (removedBook != null) {  
 books2.remove(removedBook.getbookId());  
 textArea.setText("Book Removed: " + title);  
 } else {  
 textArea.setText("Your book is not here.");  
 }  
 }