import java.sql.\*;

import java.util.Scanner;

class Book {

    private int id;

    private String title;

    private String author;

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getTitle() {

        return title;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    public String getAuthor() {

        return author;

    }

    public void setAuthor(String author) {

        this.author = author;

    }

}

class BookNotFoundException extends Exception {

    public BookNotFoundException(String message) {

        super(message);

    }

}

class BookDAO {

    private Connection connection;

    public BookDAO(Connection connection) {

        this.connection = connection;

    }

    public void insertBook(Book book) throws SQLException {

        String query = "INSERT INTO books (title, author) VALUES (?, ?)";

        try (PreparedStatement statement = connection.prepareStatement(query)) {

            statement.setString(1, book.getTitle());

            statement.setString(2, book.getAuthor());

            statement.executeUpdate();

        }

    }

    public void updateBook(Book book) throws SQLException, BookNotFoundException {

        String query = "UPDATE books SET title = ?, author = ? WHERE id = ?";

        try (PreparedStatement statement = connection.prepareStatement(query)) {

            statement.setString(1, book.getTitle());

            statement.setString(2, book.getAuthor());

            statement.setInt(3, book.getId());

            int rowsUpdated = statement.executeUpdate();

            if (rowsUpdated == 0) {

                throw new BookNotFoundException("Book not found with ID: " + book.getId());

            }

        }

    }

    public void deleteBook(int id) throws SQLException, BookNotFoundException {

        String query = "DELETE FROM books WHERE id = ?";

        try (PreparedStatement statement = connection.prepareStatement(query)) {

            statement.setInt(1, id);

            int rowsDeleted = statement.executeUpdate();

            if (rowsDeleted == 0) {

                throw new BookNotFoundException("Book not found with ID: " + id);

            }

        }

    }

    public Book getBookById(int id) throws SQLException, BookNotFoundException {

        String query = "SELECT \* FROM books WHERE id = ?";

        try (PreparedStatement statement = connection.prepareStatement(query)) {

            statement.setInt(1, id);

            try (ResultSet resultSet = statement.executeQuery()) {

                if (resultSet.next()) {

                    Book book = new Book();

                    book.setId(resultSet.getInt("id"));

                    book.setTitle(resultSet.getString("title"));

                    book.setAuthor(resultSet.getString("author"));

                    return book;

                } else {

                    throw new BookNotFoundException("Book not found with ID: " + id);

                }

            }

        }

    }

}

class BookService {

    public boolean validateBook(Book book) {

        if (book.getTitle() == null || book.getTitle().isEmpty()) {

            System.out.println("Book title is required.");

            return false;

        }

        if (book.getAuthor() == null || book.getAuthor().isEmpty()) {

            System.out.println("Book author is required.");

            return false;

        }

        return true;

    }

}

public class LibraryManagementSystem {

    public static void main(String[] args) {

        String url = "jdbc:mysql://localhost:3306/library";

        String username = "your\_username";

        String password = "your\_password";

        try {

            Connection connection = DriverManager.getConnection(url, username, password);

            BookDAO bookDAO = new BookDAO(connection);

            BookService bookService = new BookService();

            Scanner scanner = new Scanner(System.in);

            while (true) {

                System.out.println("1. Add a book");

                System.out.println("2. Update a book");

                System.out.println("3. Delete a book");

                System.out.println("4. Get a book by ID");

                System.out.println("5. Exit");

                System.out.print("Enter your choice: ");

                int choice = scanner.nextInt();

                if (choice == 1) {

                    // Add a book

                    Book book = new Book();

                    System.out.print("Enter book title: ");

                    scanner.nextLine();

                    book.setTitle(scanner.nextLine());

                    System.out.print("Enter book author: ");

                    book.setAuthor(scanner.nextLine());

                    if (bookService.validateBook(book)) {

                        bookDAO.insertBook(book);

                        System.out.println("Book added successfully.");

                    }

                } else if (choice == 2) {

                    // Update a book

                    System.out.print("Enter book ID: ");

                    int id = scanner.nextInt();

                    try {

                        Book book = bookDAO.getBookById(id);

                        System.out.print("Enter new book title: ");

                        scanner.nextLine();

                        book.setTitle(scanner.nextLine());

                        System.out.print("Enter new book author: ");

                        book.setAuthor(scanner.nextLine());

                        if (bookService.validateBook(book)) {

                            bookDAO.updateBook(book);

                            System.out.println("Book updated successfully.");

                        }

                    } catch (BookNotFoundException e) {

                        System.out.println(e.getMessage());

                    }

                } else if (choice == 3) {

                    // Delete a book

                    System.out.print("Enter book ID: ");

                    int id = scanner.nextInt();

                    try {

                        bookDAO.deleteBook(id);

                        System.out.println("Book deleted successfully.");

                    } catch (BookNotFoundException e) {

                        System.out.println(e.getMessage());

                    }

                } else if (choice == 4) {

                    // Get a book by ID

                    System.out.print("Enter book ID: ");

                    int id = scanner.nextInt();

                    try {

                        Book book = bookDAO.getBookById(id);

                        System.out.println("Book ID: " + book.getId());

                        System.out.println("Title: " + book.getTitle());

                        System.out.println("Author: " + book.getAuthor());

                    } catch (BookNotFoundException e) {

                        System.out.println(e.getMessage());

                    }

                } else if (choice == 5) {

                    // Exit the program

                    break;

                } else {

                    System.out.println("Invalid choice. Please try again.");

                }

                System.out.println();

       }

            scanner.close();

            connection.close();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}