```
#include <iostream>
#include <string>
#include <map>

using namespace std;

class Book {
public:
    int bookId;
    string title;
    string author;
    string genre;

    void displayBookDetails() const {
        cout << "Book ID: " << bookId << "\nTitle: " << title <<
"\nAuthor: " << author << "\nGenre: " << genre << endl;
    }
};

class Customer {
public:
    string name;
    string contactNumber;

    void displayCustomerDetails() const {
        cout << "Customer Name: " << name << "\nContact Number:
" << contactNumber << endl;
    }
};

class BookshopManagementSystem {
private:
}

    map<int, Book> bookInventory; // Using a map to store books
sorted by bookId

public:
    void addBook() {
        Book newBook;
        cout << "Enter Book ID: ";
        cin >> newBook.bookId;
        cin.ignore(); // Clear newline from buffer
        cout << "Enter Title: ";
        getline(cin, newBook.title);
        cout << "Enter Author: ";
        getline(cin, newBook.author);
        cout << "Enter Genre: ";
        getline(cin, newBook.genre);

        bookInventory[newBook.bookId] = newBook;
        cout << "Book added successfully!" << endl;
    }

    void displayAllBooks() const {
        if (bookInventory.empty()) {
            cout << "No books available in the inventory." << endl;
            return;
        }

        cout << "Book Inventory:\n";
        for (const auto& entry : bookInventory) {
            entry.second.displayBookDetails();
            cout << "------------------------\n";
        }
    }
    void modifyBook(int bookId) {
        if (bookInventory.find(bookId) != bookInventory.end()) {
            Book& book = bookInventory[bookId];
            cout << "Modify Book ID " << bookId << ":\n";
            cin.ignore(); // Clear newline from buffer
            cout << "Enter New Title: ";
            getline(cin, book.title);
            cout << "Enter New Author: ";
            getline(cin, book.author);
            cout << "Enter New Genre: ";
            getline(cin, book.genre);

            cout << "Book modified successfully!" << endl;
        } else {
            cout << "Book with ID " << bookId << " not found." <<
endl;
        }
    }

    void deleteBook(int bookId) {
        if (bookInventory.erase(bookId)) {
            cout << "Book with ID " << bookId << " deleted
successfully!" << endl;
        } else {
            cout << "Book with ID " << bookId << " not found." <<
endl;
        }
    }

    void searchBook(int bookId) const {
        auto it = bookInventory.find(bookId);
        if (it != bookInventory.end()) {
            cout << "Book found:\n";
            it->second.displayBookDetails();
        } else {
            cout << "Book with ID " << bookId << " not found." <<
endl;
        }
    }

    void buyBook(int bookId) {
        auto it = bookInventory.find(bookId);
        if (it != bookInventory.end()) {
            Customer customer;
            cout << "Enter customer name: ";
            cin.ignore(); // Clear newline from buffer
            getline(cin, customer.name);
            cout << "Enter contact number: ";
            getline(cin, customer.contactNumber);

            // Display the book details and customer details
            cout << "\nTransaction Details:\n";
            it->second.displayBookDetails();
            customer.displayCustomerDetails();

            // Remove the book from the inventory after purchase
            bookInventory.erase(it);
            cout << "\nBook purchased successfully!\n";
        } else {
            cout << "Book with ID " << bookId << " not found." <<
endl;
        }
    }
```

```
};

int main() {
    BookshopManagementSystem bookshop;
    char choice;

    do{
        cout << "Menu:\n1. Add Book\n2. Display All Books\n3. Modify Book\n4. Delete Book\n5. Search Book\n6. Buy Book\n7. Exit\nEnter your choice: ";
        cin >> choice;

        switch (choice) {
        case '1':
            bookshop.addBook();
            break;
        case '2':
            bookshop.displayAllBooks();
            break;
        case '3': {
            int bookId;
            cout << "Enter Book ID to modify: ";
            cin >> bookId;
            bookshop.modifyBook(bookId);
            break;
        }
        case '4': {
            int bookId;
            cout << "Enter Book ID to delete: ";
            cin >> bookId;
            bookshop.deleteBook(bookId);
            break;
        }
        case '5': {
            int bookId;
            cout << "Enter Book ID to search: ";
            cin >> bookId;
            bookshop.searchBook(bookId);
            break;
        }
        case '6': {
            int bookId;
            cout << "Enter Book ID to buy: ";
            cin >> bookId;
            bookshop.buyBook(bookId);
            break;
        }
        case '7':
            cout << "Exiting the Bookshop Management System.
Thank you!\n";
            break;
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    } while (choice != '7');

    return 0;
}
```