Lancaster's Music Hall System
RequirementsTitle:
Sharing InterfacesAuthor: Amaan Ali
Date of Publication: 19/02/25
Version Number: v3.0

1. Introduction

This document outlines the requirements for the **Operations and Box Office teams** to develop an integrated system that supports the **Marketing team's needs** at Lancaster's Music Hall. The system should enable **efficient scheduling, ticketing, group bookings, and venue coordination** while allowing seamless data sharing between teams.

The marketing team needs access different data from the operations team to optimize film scheduling, group bookings, and venue promotions. The following specification outlines the required data and its expected format.

What is an interface

An Interface in Java programming language is defined as an abstract type used to specify the behaviour of a class.
In our case we need interfaces to understand what data to access from other teams.
"You need to share the data not the database"

What we need to provide

We need to provide means by which other teams can access our data without accessing our database. We can control what other teams can
do with our data.

By week 4 we are expected to provide:

- Specification documents
  - what we need from operations
  - what we need from box office
- **Java interfaces**
  - How we are going to provide access to requested data

---

2. Requirements from Operations

2.1 Venue Calendar Data

-To identify empty slots for films, group bookings, and other events while maintaining a common venue calendar.

Data Structure:

- host_id: int
- event_id: int
- event_name: string
- venue_id: int
- venue_name: string
- date: date
- start_time: dateTime

- end_time: dateTime
- status: string

Additional Functional Requirements:

- The system should **automatically detect scheduling conflicts**.
- Venue holds should **expire after 28 days** if not confirmed, with an **automated reminder 5 days before expiration**.
- The calendar should show the **current setup condition** of rooms, including **set-up and breakdown periods**.

2.2 Seating Data

-To **allocate seats for group bookings** and ensure accessibility for patrons.

Data Structure:

- venue_id: int
- seat_id: int
- row_number: int
- seat_number: int
- status: string
- restricted_view: bool
- date: string
- start_time: string
- end_time: string

Additional Functional Requirements:

- Box Office must **review seating allocations** to ensure compliance with accessibility guidelines.
- The system should allow **Marketing to place holds on bulk seats** for group bookings and Friends of Lancaster members.

2.3 Integration with Tours & Venue Usage

-To coordinate **educational tours** and **maximize venue utilization**.

Additional Functional Requirements:

- The calendar should display **available slots** for tours and allow **Marketing to reserve them**.
- The system should track **attendee limits** to ensure **manageable group sizes**.

---

3. Requirements from Box Office

3.1 Ticket Sales & Seat Management

-To **synchronize ticket sales across multiple platforms** while maintaining accurate seating records.
Access Level: Real-time access with modification rights for ticketing.

Data Structure:

- booking_id: int
- customer_id: int
- event_id: int
- price: int
- quantity: int
- booking_date: string
- discount_type: string

Additional Functional Requirements:

- Reserved seating areas (VIP, Friends of Lancaster) should be clearly marked.
- The system should **automatically block restricted-view seats** and ensure **accessible seating allocation**.
- Holds on tickets should **expire after a set period** (e.g., **48 hours for standard holds, 7 days for group bookings**) with **automated reminders sent to customers**.

3.2 Group Bookings & Special Reservations

-To handle **large group reservations efficiently**.

Additional Functional Requirements:

- Group bookings (12+ people) should be flagged for **Operations to ensure logistical readiness**.
- The system should support **bulk pricing and seating adjustments** for large groups.
- Special ticket types (e.g., **complimentary tickets, artist guest lists**) should be managed with tracking and approval.

3.3 Customer Data Access

-To support **personalized communication** for "Friends of Lancaster" members.

Data Structure:

- customer_id: int
- customer_name: string
- email: string
- phone_number: int
- friend_member: bool

Additional Functional Requirements:

- The system should allow **Marketing to track repeat customers** for targeted campaigns.
- Customer support should have access to **modify reservations, process refunds, and reissue tickets** with proper logging.

3.4 Integration with Marketing

- To **track sales trends and support targeted promotions**.
Access Level: Read-only for Marketing, full access for Box Office.

Additional Functional Requirements:

- The system should track **ticket sales and provide analytics** for Marketing.
- Marketing should be able to schedule **discounts and bundled ticket offers**.
- The system should allow Marketing to **place seat holds** for Friends of Lancaster, ensuring priority access.