**Case Study: Online Ticket Booking System**

**Context:**

You are required to develop a console-based Online Ticket Booking System using Core Java. The application will allow users to book tickets for various events, manage bookings, and perform various business operations related to ticket management. The application will utilize Java Collections for storing event and booking information, implement Object-Oriented Programming principles, and include exception handling to manage potential errors.

**Requirements:**

**Features:**

1. **View Events:**
   * Allow users to view a list of available events.
   * Display event details such as name, date, venue, and ticket availability.
2. **Book Tickets:**
   * Allow users to book tickets for events.
   * Display booking confirmation with details such as event name, ticket price, and total cost.
3. **Cancel Booking:**
   * Allow users to cancel a booking.
   * Update ticket availability accordingly.
4. **View Bookings:**
   * Allow users to view their booking history.
   * Display booking details such as event name, date, and ticket price.

**Classes and Objects:**

1. **Event Class:**
   * Attributes: id, name, date, venue, ticketPrice, availableTickets.
   * Methods: Constructors, getters and setters, toString method.
2. **Booking Class:**
   * Attributes: id, eventId, userId, bookingDate, totalCost.
   * Methods: Constructors, getters and setters, toString method.
3. **TicketBookingSystem Class:**
   * Attributes: ArrayList<Event> events, ArrayList<Booking> bookings.
   * Methods:
     + viewEvents(): Displays a list of available events.
     + bookTicket(int eventId, int userId, int quantity): Books tickets for an event.
     + cancelBooking(int bookingId): Cancels a booking.
     + viewBookings(int userId): Displays booking history for a user.
     + Helper methods for input validation and exception handling.

**Deliverables:**

1. Complete source code for the Online Ticket Booking System.
2. Documentation including:
   * How to run the application.
   * Instructions for each feature.
   * Explanation of the exception handling implemented.
3. A brief report on the application design and how Object-Oriented principles were applied.