**Domain Description:**

We’re building a system to manage events, such as conferences, concerts, workshops, or sports tournaments. Here are the key components:

**Event**: Represents a specific happening, like a conference or a music festival.

**Organizer**: The entity responsible for planning and executing the event.

**Attendee**: Individuals who participate in the event.

**Ticket**: Provides access to the event for attendees.

**Venue**: The physical location where the event takes place.

**Schedule**: The timeline or agenda of the event, including sessions, breaks, and activities.

**Sponsor**: Companies or organizations that support the event financially or through other means.

**Budget**: The financial plan for organizing the event.

**Promotion**: Marketing and advertising efforts to attract attendees.

**Feedback**: Input from attendees or organizers to improve future events.

**Entity-Relationship (ER) Diagram:**

Let’s represent these entities and their relationships visually using an ER diagram.

**Entities**:

**Event**: EventID (Primary Key), Name, Date, Description

**Organizer**: OrganizerID (Primary Key), Name, Contact Info

**Attendee**: AttendeeID (Primary Key), Name, Email

**Ticket**: TicketID (Primary Key), EventID (Foreign Key), Price, Type

**Venue**: VenueID (Primary Key), Name, Address

**Schedule**: ScheduleID (Primary Key), EventID (Foreign Key), TimeSlot, Activity

**Sponsor**: SponsorID (Primary Key), Name, Contribution

**Budget**: BudgetID (Primary Key), EventID (Foreign Key), ExpenseCategory, Amount

**Promotion**: PromotionID (Primary Key), EventID (Foreign Key), Channel, Content

**Feedback**: FeedbackID (Primary Key), EventID (Foreign Key), AttendeeID (Foreign Key), Rating, Comments

**Classes and Attributes**

**1.Event:**

**Attributes:**

-Eventid

-Name

-Date

-Description

**Methods**:

+CreateEvent()

+UpdateEvent()

+DeleteEvent()

**2.Organizer**

**Attributes:**

-OrganizerId()

-Name

-ContactInfo

**Methods**:

+AddOrganizer()

+UpdateOrganizer()

+RemoveOrganizer()

**3.Attendee**

**Attributes:**

-AttendeeId

-Name

-Email

**Methods**:

+RegisterAttendee()

+checkInAttendee()

+SendConfirmationEmail()

**Ticket**

**Attributes:**

-TicketID

-EventID

-Price

-Type

**Methods**:

+GenerateTicket()

+SellTicket()

+ValidateTicket()

**Venue**

**Attributes:**

-VenueID

-Name

-Address

**Methods**:

+ReserveVenue()

+UpdateVenueDetails()

+CancelVenueBooking()

**Schedule**

**Attributes:**

-ScheduleID

-EventID

-TimeSlot

-Activity

**Methods**:

+CreateSchedule()

+UpdateSchedule()

+DeleteSchedule()

**Sponsor**

**Attributes:**

-SponsorID

-Name

-Contribution

**Methods**:

+AddSponsor()

+UpdateSponsorContribution()

+RemoveSponsor()

**Budget**

**Attributes:**

-BudgetID

-EventID

-ExpenseCategory

-Amount

**Methods**:

+TrackExpenses()

+AllocateBudget()

+GenerateExpenseReport()

**Promotion**

**Attributes:**

-PromotionID

-EventID

-Channel

-Content

**Methods**:

+PromoteEvent()

+MonitorPromotionMetrics()

+AdjustPromotionStrategy()

**Feedback**

**Attributes:**

-FeedbackID

-EventID

-AttendeeID

-Rating

-Comments

**Methods**:

+CollectFeedback()

+AnalyzeFeedback()

+ImproveEventExperience()

**Relationships**:

An Event is organized by an Organizer (one-to-many).

An Event has multiple Attendees (many-to-many).

An Event has one or more Tickets available (one-to-many).

An Event takes place at a specific Venue (many-to-one).

An Event has a detailed Schedule (one-to-many).

An Event may have multiple Sponsors (many-to-many).

An Event has a Budget (one-to-one).

An Event is promoted through various Promotion channels (one-to-many).

Attendees provide Feedback for an Event (many-to-many).

**Uml** **Diagram**:

