

Final Project Report

Poorva Joshi

Northeastern University

ITC6000: Database Management Systems

Professor Sergey Shevchenko

Oct 22, 2023

Introduction

Project name: Event-Based Management

I have chosen to embark on the development of an event management system, recognizing the significance of corporate events within the realm of business growth. This concept has piqued our interest, as such a system is designed to assist event organizers in the seamless organization, execution, and coordination of corporate events, encompassing conferences, job expos, seminars, meetings, and the like. This web-based application allows event attendees to register for and attend these events efficiently. Additionally, it empowers companies to strategically engage external stakeholders by harnessing insights derived from data collected during previous events.

The application comprises the following key features:

- Online registration.
- Ticketing services.
- Event marketing tools.
- Event logistics management capabilities.
- Participant outreach functions.
- Post-event data management.

The application caters to two primary user groups: event attendees and event organizers or host companies. Event organizers utilize the software for various phases, including event planning, promotion, and post-event activities. Attendees, on the other hand, use the application to register, obtain tickets, access essential event information such as schedules and locations, and establish connections for post-event networking. Organizations subscribing to the application can choose between event-specific contracts or longer-term agreements, such as annual or biannual plans. It's important to note that the application is provided free of charge to event attendees.

Business Analysis

The application serves two primary user groups: event attendees and event hosts, which may be individual organizers or companies. Attendees utilize the application for tasks such as event registration, ticket retrieval, accessing essential event details, and establishing post-event connections for networking purposes. On the other hand, event hosts benefit from the application in the following ways:

- Planning and configuring events.
- Scheduling guest speakers.

- Creating virtual live-streamed sessions.
- Collecting attendee registration data.
- Formulating marketing campaigns to engage event participants.

Monitoring performance metrics and conducting in-depth analysis of event-related data. Major data entities include:

- Attendee information provided at registration,
- Session information,
- Speaker information,
- Location/Conference Room,
- and Event Staff.

Attendee contact information will be a crucial data point for pre-event marketing material and post-event follow-up. Additionally, session, speaker, and location attributes will be an important part of planning the overall event schedule. Only one session can occur in a conference room at a specific time and a speaker only conducts one session at a time, but a speaker may host many sessions during the conference. Post-event, the host can track attendance numbers compared to registration, which sessions and speakers were most popular, analyze attendee demographics, and send out satisfaction surveys.

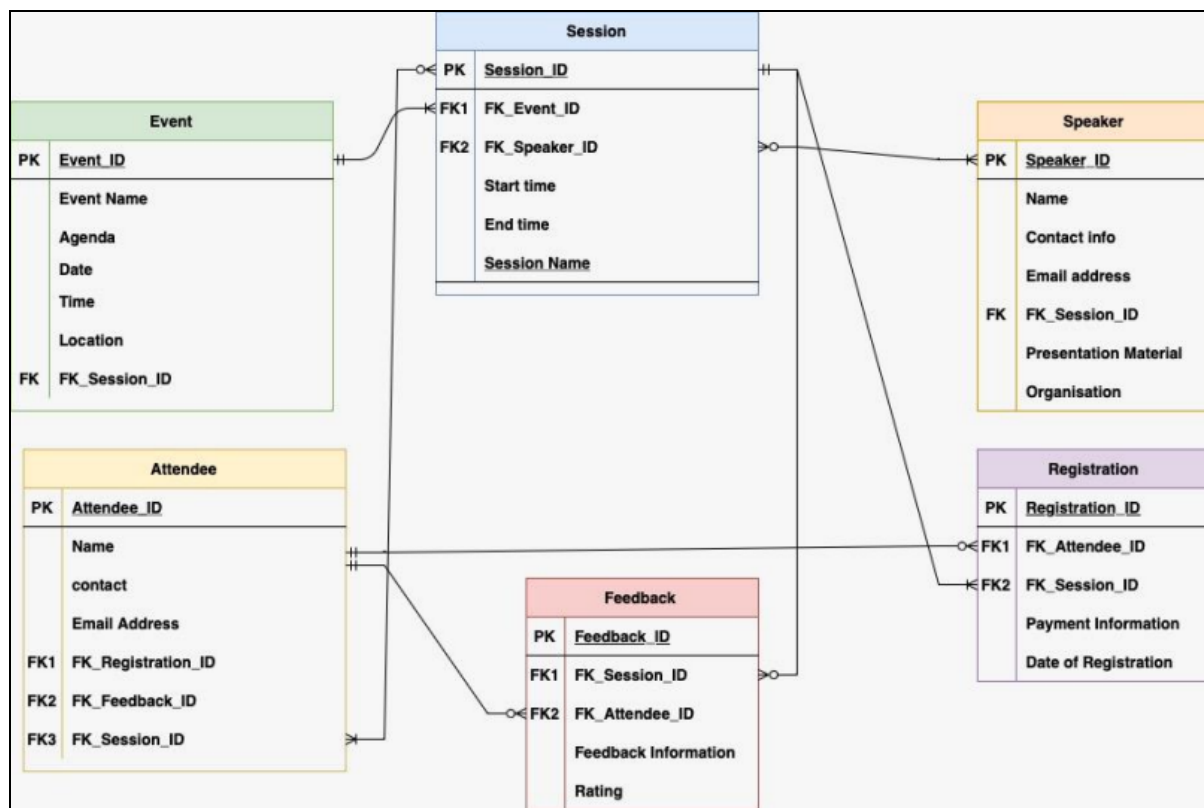
Attendees can use the application to:

- Register
- Retrieve tickets
- Review conference schedule
- RSVP for sessions
- Network with other conference attendees

Attendees can utilize the app during the conference to RSVP for sessions, stay on schedule, pre-plan their day, and connect with other conference attendees in real-time. Post-conference, attendees can submit feedback for previously attended conferences and receive notifications about upcoming conferences based on their preferences. Attendees can be fully engaged with the app during the conference, or they can use just the ticketing and registration functions.

Table Design and Analysis

Entity Relationship Diagram:-



1. **Attendees (One-to-Many with Registrations):** Attendees have the ability to register for multiple events and sessions, but each registration is linked to a single attendee. This creates a one-to-many relationship between Attendees and Registrations.
2. **Sessions (One-to-Many with Registrations):** Each session can be associated with multiple registrations from various attendees, yet each registration is specific to a single session. This results in a one-to-many relationship between Sessions and Registrations.
3. **Events (One-to-Many with Sessions):** Each event has the capacity to encompass multiple sessions, with each session being linked to a single event. This establishes a one-to-many relationship between Events and Sessions.
4. **Sessions (Many-to-One with Events):** Each session is connected to a single event, while an event can host multiple sessions. This defines a many-to-one relationship between Sessions and Events.

5. **Speakers (Many-to-Many with Sessions):** Speakers can be involved in multiple sessions, and each session can feature multiple speakers. This dynamic interaction forms a many-to-many relationship, facilitated by the Session_ID within the Speakers table.
6. **Feedback (Many-to-One with Attendees, Events, and Sessions):** Each piece of feedback is linked to a single attendee, a specific event, and a particular session. Nevertheless, a single attendee can offer feedback for multiple events and sessions, signifying a many-to-one relationship between Feedback and Attendees, Events, and Sessions.
7. **Registrations (Many-to-One with Attendees and Events):** Each registration is connected to a single attendee and a specific event. Nonetheless, a single attendee can possess multiple registrations for various events, forming a many-to-one relationship between Registrations and Attendees and between Registrations and Events.

Database Implementation

The SQL command provided below retrieves information about attendees who have given a 5-star rating for a particular session. Event organizers utilize this data to gain insights into which speakers and sessions were particularly well-received and popular among attendees.

Action: Retrieve Session Name, Speaker Name, and Attendee Name for Sessions with Attendees who provided a Feedback Rating of 5:

```
SELECT S.Name AS "Session Name", SP.Name AS "Speaker Name", A.Name
AS "Attendee Name"
FROM Session S
JOIN Feedback F ON S.Session_ID = F.Session_ID
JOIN Speaker SP ON S.Speaker_ID = SP.Speaker_ID
JOIN Attendee A ON F.Attendee_ID = A.Attendee_ID
WHERE F.Rating = 5;
```

	Session Name	Speaker Name	Attendee Name
1	Coop Program	Robert Davis	Naresh Kumar
2	Welcome Address	John Smith	Lauren Ogburn
3	Panel Discussion	Sarah Wilson	Praveen
4	Panel Discussion	Sarah Wilson	Poorva Joshi
5	Welcome Address	John Smith	Praveen
6	Coop Program	Robert Davis	Emma
7	Networking Session	Mary Johnson	Hazique

The following use case demonstrates the calculation of the average rating provided by an attendee. This data can be valuable in assessing overall attendee satisfaction and identifying whether most attendees had a positive response to the majority of the sessions.

Action: List of Attendees and the Average Rating of their Feedback:

```
SELECT A.Name AS "Attendee Name", AVG(F.Rating) AS "Average Rating"
FROM Attendee A
JOIN Feedback F ON A.Attendee_ID = F.Attendee_ID
GROUP BY A.Name;
```

	Attendee Name	Average Rating
1	Lauren Ogburn	5.0
2	Poorva Joshi	5.0
3	Praveen	5.0
4	Arjun Reddy	4.85
5	Emma	5.0
6	Hazique	5.0
7	Naresh Kumar	5.0
8	Tanvi Sunil	4.7

The SQL command below reveals the correlation between specific events, the corresponding speakers, and the dates on which these events were conducted. Organizers find this information valuable for keeping track of upcoming events and for the timely distribution of feedback forms once a particular event has concluded.

Action: List Speaker Name, Session hosted and Event Date:

```
SELECT SP.Name AS "Speaker Name", S.Name AS "name", E."Date" AS
"Event Date"
FROM Session S
JOIN Speaker SP ON S.Speaker_ID = SP.Speaker_ID
JOIN Event E ON S.Event_ID = E.Event_ID
JOIN Registration R ON S.Session_ID = R.Session_ID
ORDER BY E.date ASC;
```

	Speaker Name	name	Event Date
1	John Smith	Welcome Address	2023-04-10
2	John Smith	Welcome Address	2023-04-10
3	John Smith	Welcome Address	2023-04-10
4	Robert Davis	Coop Program	2023-04-15
5	Robert Davis	Coop Program	2023-04-15
6	Sarah Wilson	Panel Discussion	2023-04-20
7	Sarah Wilson	Panel Discussion	2023-04-20
8	Sarah Wilson	Panel Discussion	2023-04-20
9	Mary Johnson	Networking Session	2023-04-20
10	Mary Johnson	Networking Session	2023-04-20

Analytics, Reports, and Metrics

Attendee Analysis:

Report: End-users might want to see the attendance statistics of different sessions

SQL Statement: Calculating the number of attendees for each session

```
SELECT S.Name AS Session_Name, COUNT(R.Attendee_ID) AS
Attendee_Count
FROM Session S
LEFT JOIN Registration R ON S.Session_ID = R.Session_ID
GROUP BY S.Session_ID;
```

Welcome Address	3
Panel Discussion	3
Coop Program	2
Networking Session	2

Feedback Analysis:

Report: End-Users might want to know the feedback ratings of each session

SQL Statement: Calculating the average rating for each session

```
SELECT S.Name AS Session_Name, AVG(F.Rating) AS Avg_Rating
FROM Session S
```

```
LEFT JOIN Feedback F ON S.Session_ID = F.Session_ID
GROUP BY S.Session_ID;
```

Welcome Address	4.933333333333333
Panel Discussion	4.966666666666667
Coop Program	5.0
Networking Session	4.85

Registration Trends:

Report: System administrators might want to see the registration trends over time

SQL Statement: Counting the number of registrations over time.

```
SELECT DATE(Registration_Date) AS Registration_Date,
COUNT(Registration_ID) AS Registration_Count
FROM Registration
GROUP BY DATE(Registration_Date)
ORDER BY Registration_Date;
```

2023-04-01	2
2023-04-07	1
2023-04-10	3
2023-04-11	1
2023-04-13	1
2023-04-15	1
2023-04-17	1

Speaker Performance:

Report: Evaluating the feedback ratings of different speakers

SQL Statement: Calculating the average ratings for each speaker

```
SELECT SP.Name AS Speaker_Name, AVG(F.Rating) AS Avg_Rating
FROM Speaker SP
JOIN Session S ON SP.Speaker_ID = S.Speaker_ID
LEFT JOIN Feedback F ON S.Session_ID = F.Session_ID
GROUP BY SP.Speaker_ID;
```


John Smith	4.933333333333333
Mary Johnson	4.85
Robert Davis	5.0
Sarah Wilson	4.966666666666667

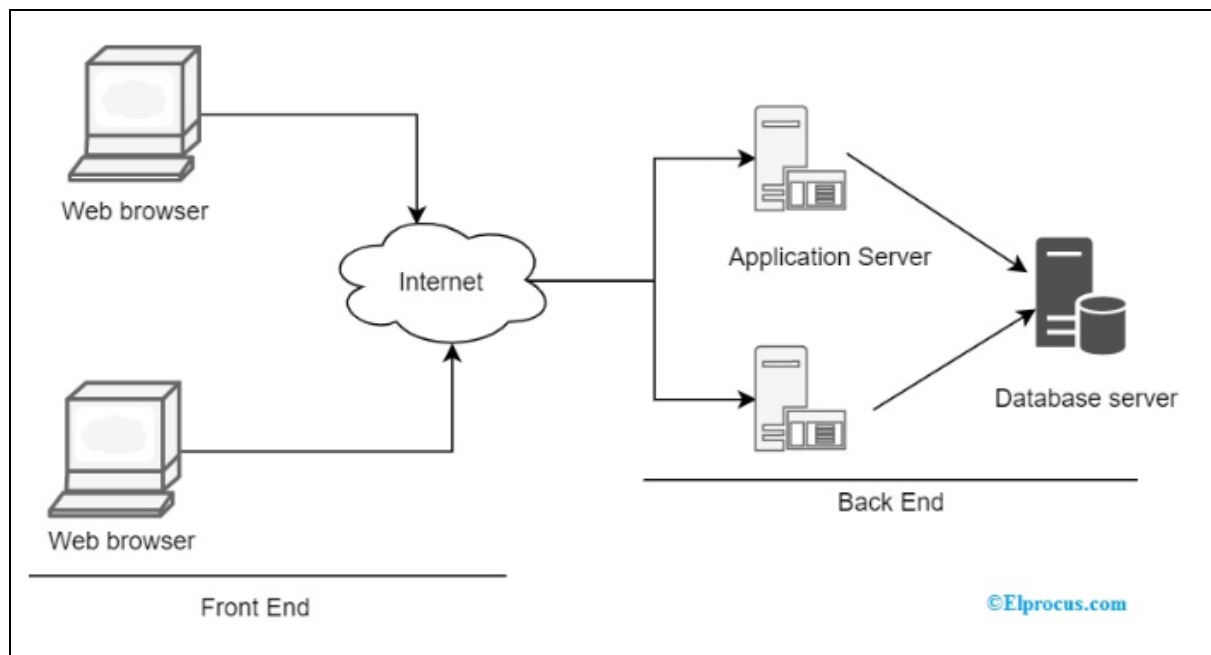
Security Concerns

1. Demographic Information Privacy:
 - a) Demographic information about attendees, such as age, gender, or ethnicity, is considered sensitive data.
 - b) It should not be disclosed or mentioned on the event's tickets, as this information could potentially lead to privacy concerns or misuse.
2. Secure Payment Handling:
 - a) Payment information, including credit card details, is highly sensitive and should be handled with the utmost security.
 - b) It should be processed through a secure payment platform that complies with relevant security standards, such as PCI DSS (Payment Card Industry Data Security Standard).
 - c) Storing this payment data should be avoided to minimize the risk of data breaches and protect the financial information of attendees.
3. Restricted Attendee Contact Information:
 - a) Attendee contact information, including email addresses and physical addresses, should be treated with care.
 - b) It should only be shared within the organizing organization or the event management team, not publicly displayed on the attendees list. This ensures that attendees' contact details are not exposed to external parties or potentially misused.
4. Controlled Sharing of Speakers' Contact Information:
 - a) Speakers' contact information is typically intended for official event use.
 - b) It should only be shared with authorized business users, such as event organizers or staff responsible for event logistics.

- c) It should not be made accessible to all attendees to maintain speaker privacy and prevent unsolicited communications.

By adhering to these guidelines, event organizers can enhance data privacy and security, safeguard sensitive information, and create a more secure and trustworthy event experience for all participants.

Architecture



The Solution Client/Server Architecture: Our database system will follow a client/server architecture model. The client and server work together to provide a seamless and efficient user experience. The client sends requests to the server, which processes these requests and returns the appropriate results or data to the client. This architecture provides the foundation for efficient data management, secure access, and scalable performance while enabling users to access and manipulate data from anywhere, enhancing the application's accessibility and usability.

Cloud Hosting: Both the application and the database will be hosted in the cloud. This approach offers several advantages, including simplicity, scalability, accessibility, and reliability. We prefer cloud hosting to on-premises since it is often easier in terms of initial setup and management because cloud providers handle much of the underlying infrastructure.

Storage Requirements: Our database will include six tables namely Event, Session, Attendee, Speaker, Registration and Feedback. The volume of data stored in the tables is expected to grow over time as more events, speakers, sessions, attendees, registrations, and feedback are recorded. Hence,

the database system must be designed to scale horizontally or vertically as data volume and user activity increase. This will be essential for accommodating future growth. Data integrity will be a primary concern, ensuring that each record in the database is accurate and reliable. This will be achieved through the implementation of proper data validation and constraints. Security measures will be implemented to protect sensitive information, including attendee contact details and payment information. A strategy for seamless data migration will be developed, ensuring that the transition to newer storage technologies or providers can be achieved without data loss or service interruptions.

Project Wrap-up and Future Considerations

Long-term crucial planning: Consider things like system updates, data migrations, and developing technologies when making long-term plans.

Collaboration: The success of the project depends on collaboration between IT teams, business divisions, and stakeholders.

User-Centric Design: One of the most important lessons is how to design intuitive user interfaces and experiences for administrators as well as end users. Higher adoption rates and better administration can result from a well-designed system.

Real-time monitoring and analytics: Real-time monitoring and analytics are frequently needed for event-based management to derive insights from event data. Setting up efficient monitoring and analytics tools for decision support may be learned from this.

Data modeling for events: In-depth knowledge of data modeling, particularly for events, is necessary for these tasks. The lessons learned frequently include the handling of interactions between various event kinds, the definition of event schemas, and the organization of event data.

Key Takeaways

Scalability: Scalability must be considered in the design of event-based systems. Data volume is only one aspect of scalability; another is the capacity to grow horizontally to support more users and events.

Security in Event processing: In event-based systems, security is a crucial factor. The most important takeaway is knowing how to safeguard event data and the procedures that are sparked by events. This incorporates audit trails, encryption, and access control.

Compliance Awareness: Understanding legal and regulatory compliance is crucial to these initiatives, especially about data handling and privacy.

Continuous Improvement and Adaptation: The value of consistently asking for input, keeping track of performance, and making incremental adjustments is among the most important lessons learned. With shifting business requirements and technological advancements, event-based systems must adapt.

Business Alignment: Realizing the value of DBMS alignment with organizational strategy and its contribution to business success.

Budgeting: Allocate enough resources to support the long-term upkeep and development of your DBMS. This covers the cost of hiring staff, purchasing new hardware, and software licenses.

References/Citations

5. Makadiya, M. (2013). Making a Business Case for Event Management Software. *Business Analyst Learnings*. September 24, 2023, <https://www.businessanalystlearnings.com/technology-matters/2013/8/14/making-a-business-case-for-event-management-software>
6. Mohan, R. (2022). Event Management System: Everything you need to know. *Eventzilla*. September 24, 2023, <https://www.eventzilla.net/blog/event-resources/event-management-system-everything-you-need-to-know/#:~:text=An%20EMS%20can%20save%20you,self%2Dregister%20for%20your%20event>
7. <https://vertabelo.com/blog/how-to-plan-and-run-events-an-event-management-data-model/>
8. <https://youtu.be/lAtCySGDD48?si=EeWFFMQ2QeFj5ZR1>
9. Emil,D.(2017). [An Event Management Data Model | Vertabelo Database Modeler](#)
10. Daniel,S.(2019). [Creating Entity Relationship Diagrams using Draw.io - YouTube](#)