

USE CASE STUDY REPORT

Group No.: Group 13

Student Names: Pavithra Moorthy & Rachita Anand

Executive Summary:

The Objective of this study case is to implement a website to organize an event. Due to Covid-19 restrictions, virtual hosting has become a trend and people who can't attend the event in-person for various reasons, can attend virtually. Part of planning the event also means planning the budget, determining what type of event and fixing the event date. This system helps to identify planning committee, program outline and the planning process to achieve event goals for web-based methods. To confirm the venue and performers reservations and identify the constituency for the guest list and check the dates. Discussing the invitation list and estimating the yield and then confirming invitation lists for events and meals with necessary employees. Design and develop event database tracking RSVP's and mailing invitations.

The system allows the user to login through its first page and Customers use the system to make booking and schedule events in the day they want. Event Management system is manual and only accessible to staff. The client must travel to the company offices in order to schedule, book and organize an event such as Birthday Party, Wedding, Charity, Banquets, etc. Clients pay to book for an event which is inconveniencing when customers are many at the company. It takes lots of time of customer because they must search such event organizer and contact them individually so an online event management system is needed which will enable the customer make booking, schedule events at any preferred time.

I. Introduction

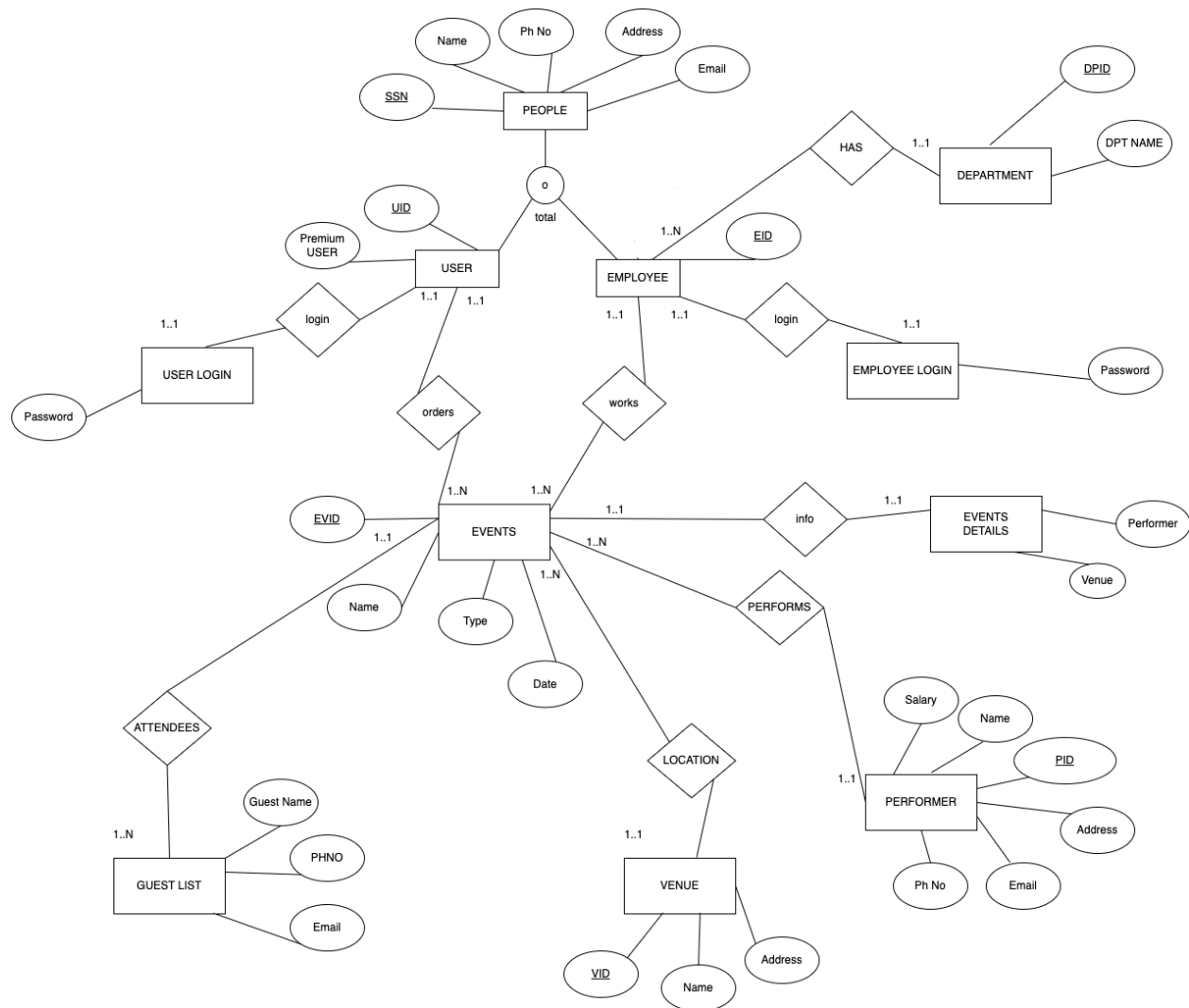
Hybrid event platform is a web-based application that enhanced project management to the creation and development of large-scale events. It involved studying the brand, identifying its target audience, devising the event concept, and coordinating the technical aspects before launching the event. The Event management system enables customers to view various packages about the event and make booking through the online platform. The process of planning and coordinating the event is usually referred to as event planning and it included budgeting, date, venue, acquiring necessary permits, coordinating transportation, performers, arranging decorations, catering, coordinating with third party vendors, and emergency plans. Each event was different in its nature so process of planning & execution of each event differed on basis of type of event. Thus, creating ample time for the customers to manage and schedule events.

A person can have only one account and must have only one login credentials. An event is managed by only one event coordinator. The event coordinator allots venue, performer and other tasks according to the customer requirements. The event coordinator gives tasks to the required departments depending on what type of event it is. The performers are not required for all types of events, they only required for some events. If an event is held offline then a venue is required, if

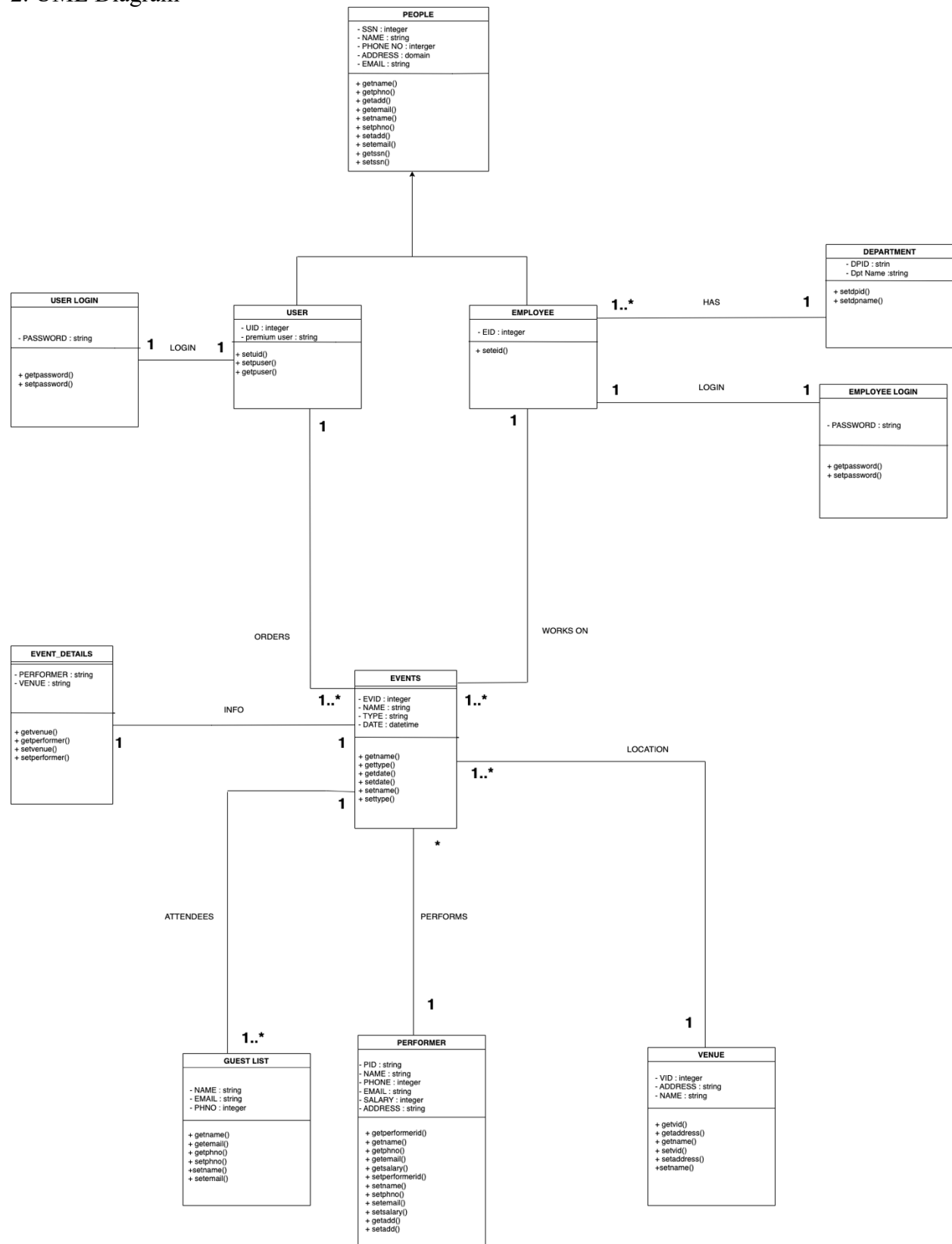
the event is held online venue is not necessary. An event can have any number of guests. A customer can order any number of events.

II. Conceptual Data Modeling

1. EER Diagram



2. UML Diagram



III. Mapping Conceptual Model to Relational Model

Primary Key-

Foreign Key-

- PEOPLE (**SSN**, Name, PhNo, Address, Email)

- DEPARTMENT (**DPID**, DPT_NAME)

- USER (**UID**, Premium User, *SSN*)

FOREIGN KEY SSN refers to SSN in People; NULL NOT ALLOWED

- EMPLOYEE (**EID**, *DEPARTMENT*, *SSN*)

FOREIGN KEY SSN refers to SSN in People; NULL NOT ALLOWED

FOREIGN KEY DEPARTMENT refers to DPID in Department; NULL NOT ALLOWED

- USER_LOGIN (*UID*, Password)

FOREIGN KEY UID refers to UID in User; NULL NOT ALLOWED

- EMPLOYEE_LOGIN (*EID*, Password)

FOREIGN KEY UID refers to UID in User; NULL NOT ALLOWED

- VENUE (**VID**, Name, Location)

- PERFORMER (**PID**, Name, PhNo, Email, Address, Salary)

- EVENTS (**EVID**, Name, Type, *UID*, *EID*, *VID*, *PID*)

FOREIGN KEY UID refers to UID in User; NULL NOT ALLOWED

FOREIGN KEY EID refers to EID in Employees; NULL NOT ALLOWED

FOREIGN KEY VID refers to VID in Venue; NULL NOT ALLOWED

FOREIGN KEY PID refers to PID in Performer; NULL ALLOWED

- EVENT_DETAILS (*EVID*, Venue, Performer)

FOREIGN KEY EVID refers to EVID in Events; NULL NOT ALLOWED

- GUESTLIST (*EVID*, Name, PhNo, Email)

FOREIGN KEY EVID refers to EVID in Events; NULL NOT ALLOWED

IV. Implementation of Relation Model via MySQL and NoSQL

MySQL Implementation:

The database was created in MySQL and the following queries were performed:

Query 1: Find event name, event type, event date which is coming in the upcoming days

```
select name, type, date from events where date >= curdate();
```

	name	type	date
▶	Alice	banquet	2022-12-12 00:00:00
	Jacey	birthday	2022-12-30 00:00:00
	Flo	wedding	2023-01-01 00:00:00
	Dixie	banquet	2023-01-02 00:00:00
	Jordan	charity	2023-01-12 00:00:00
	Tera	workshop	2023-02-18 00:00:00
	Zen	charity	2023-03-09 00:00:00
	Lia	birthday	2023-03-22 00:00:00
	Topaz	wedding	2023-04-04 00:00:00

Query 2: Find the number of events ordered by each user

```
select uid, count(evid) as event_count from events group by uid order by count(evid);
```

	uid	event_count
▶	56	20
	99	15
	52	10
	51	8
	53	5
	54	4
	55	2
	58	1
	59	1

Query 3: Find the performer id and performer name of the top three highest salary paid to a performer

```
select pid,name, salary from performer p1 where 3>(select count(distinct pid) from performer p2 where p1.salary<p2.salary) order by salary desc;
```

	pid	name	salary	
▶	230	Jazmyn	34000	
	228	Kent	32000	
	222	Hal	30000	

Query 4: Find the number of guests for each event name and location

```
with guest_event as
(select evid,count(evid) as guest_count
from guestlist group by evid)
```

```
select g.evid, g.guest_count, e.name, v.name
from guest_event as g
inner join events as e on g.evid=e.evid
inner join venue as v on v.vid=e.vid;
```

Query 5: Find the event id, name of the events happening in Richards Hall

```
select e.evid,e.name from events e where e.vid=(select vid from venue where name='Richards
hall');
```

	evid	name	
▶	326	Madison	

Query 6: Find the department id and department name where the count of employees is greater than 10 using exists

```
with dpt_employee as
(select department, count(eid) as employee_count from employee group by department)
```

```
SELECT dpid, dpt_name
FROM department d
WHERE EXISTS (SELECT department FROM dpt_employee de WHERE d.dpid =
de.department AND employee_count > 10);
```

	dpid	dpt_name
►	MNG	Manager

Query 7 : Find the names of the premium users who ordered the events

```
select p.name, e.uid, e.type from events e, people p, user u where u.ssn=p.ssn and u.uid=e.uid and u.premium='y';
```

	name	uid	type
►	Nance	57	charity
	Darleen	58	wedding
	Callahan	59	wedding
	Keri	60	charity
	Shayne	61	baby shower
	Xanthia	65	charity
	George	66	birthday
	Merrill	67	birthday
	Shana	68	wedding

NoSQL Implementation:

The tables are created in MongoDB. The following NoSQL queries were done:

Query 1: Find the performers whose salary is greater than 1000

```
db.performer.find( {"salary": {$gt: 1000}} );
```

Result

```
{ "_id" : ObjectId("6393b83faf6dac545711a296"), "pid" : 201, "name" : "Kathy", "phno" : 8123456710 }
{ "_id" : ObjectId("6393b83faf6dac545711a297"), "pid" : 202, "name" : "Gytha", "phno" : 8123456711 }
{ "_id" : ObjectId("6393b83faf6dac545711a298"), "pid" : 203, "name" : "Aleah", "phno" : 8123456712 }
{ "_id" : ObjectId("6393b83faf6dac545711a299"), "pid" : 204, "name" : "Marvyn", "phno" : 812345671 }
{ "_id" : ObjectId("6393b83faf6dac545711a29a"), "pid" : 205, "name" : "Kevyn", "phno" : 8123456714 }
```

Query 2: Find the total employees working in each department

```
db.employee.aggregate( { "$group": { "_id": "$department", total_employees: { $sum:1 } } } );
```

Result

```
{ "_id" : "MNG", "total_employees" : 2 }
{ "_id" : "TRC", "total_employees" : 1 }
{ "_id" : "PC", "total_employees" : 1 }
{ "_id" : "HR", "total_employees" : 1 }
```

Query 3: Find the total events ordered by each user in descending order

```
db.event.aggregate( { "$group": { _id: "$uid", total_events_ordered: { $sum:1} } }, {"$sort":{"total_events_ordered":-1}});
```

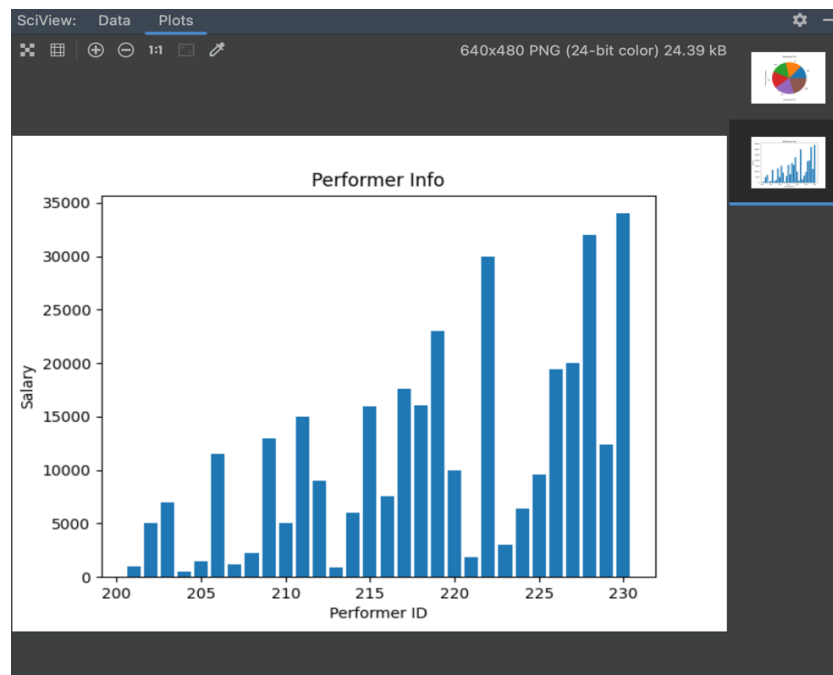
Result

```
{ "_id" : 51, "total_events_ordered" : 4 }
{ "_id" : 50, "total_events_ordered" : 3 }
{ "_id" : 54, "total_events_ordered" : 1 }
{ "_id" : 52, "total_events_ordered" : 1 }
{ "_id" : 53, "total_events_ordered" : 1 }
```

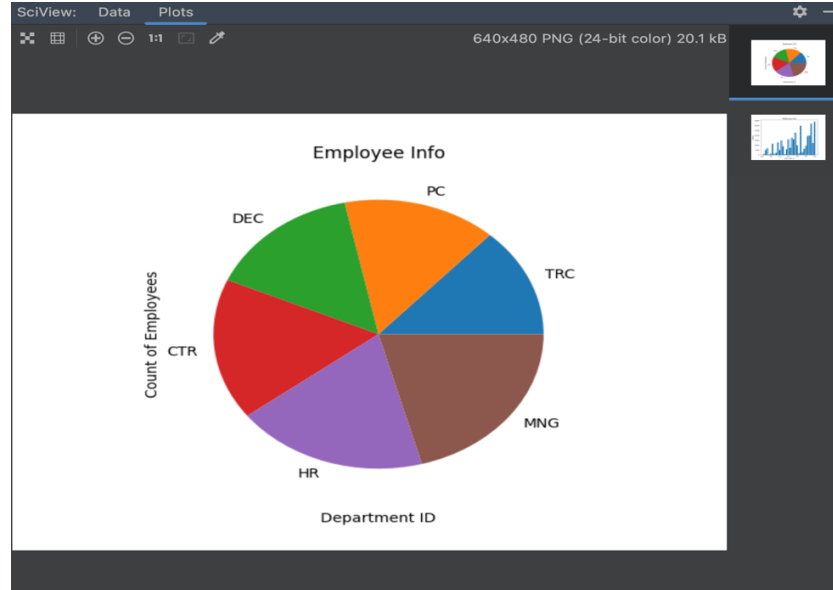
V. Database Access via Python

The database is connected to Python using MySQL connector followed by a cursor to execute the queries and fetch records using execute. The list and matplotlib are used to store data and show it in a pie chart and bar chart as follows.

GRAPH 1: The salary of each performer in a bar chart



GRAPH 2: THE COUNT OF EMPLOYEES IN EACH DEPARTMENT IN PIE CHART



OUTPUT OF MYSQL CONNECTED WITH PYTHON:

```

/Users/pavithra/PycharmProjects/pythonProject/venv/bin/python /Users/pavithra/PycharmProjects/pythonProject/main.py
Connected to MySQL Server version 8.0.30
Your connected to database: ('project',)
Query 1: Find event name, event type, event date which is coming in the upcoming days

('Alice', 'banquet', datetime.datetime(2022, 12, 12, 0, 0))
('Jacey', 'birthday', datetime.datetime(2022, 12, 30, 0, 0))
('Flo', 'wedding', datetime.datetime(2023, 1, 1, 0, 0))
('Dixie', 'banquet', datetime.datetime(2023, 1, 2, 0, 0))
('Jordan', 'charity', datetime.datetime(2023, 1, 12, 0, 0))
('Tera', 'workshop', datetime.datetime(2023, 2, 18, 0, 0))

Query 2: Find the performer id and performer name of the top three highest salary paid to a performer
(230, 'Jazmyn', 34000)
(228, 'Kent', 32000)
(222, 'Hal', 30000)

Query 3: Find the event id, name of the events happening in Richards Hall
(326, 'Madison')
next
MySQL connection is closed

Process finished with exit code 0

```

VII. Summary and Recommendation

The Event Management Platform is designed on MySQL. This case study is set out to design and develop an event management platform that would help in carrying out online event scheduling. For reliability, effectiveness, efficiency to be realized, the company will need to adopt the developed system. This study focuses on modeling a representative ‘Event Management System’ for an ‘Hybrid Event Platform’. A customer requests the company for an event management. The customer provides the details of the event and its requirements. The customers explain its aims, when and where the event will take place, how long it will last, its format, expected number of guests, venues and performers required, and other facilities required. The event coordinator studies the requirements of the event carefully and using the event management system finds the estimated cost and informs the customer about it. Successful implementation of the application concludes that SQL and NoSQL queries are implemented in python can help in evolving efficient, cost-effective, and quality Hybrid event platform

Companies should consider investing in information technology platform for easy monitoring and transactions processing. This is because the use of business information systems can help in reducing the cost of doing business and improve on the decision-making process since information that supports decision making can be accessed in a faster and timely way.