VIETNAM NATIONAL UNIVERSITY, HANOI

INTERNATIONAL SCHOOL

FINAL REPORT

Enterprise Information Systems

Topic: Building an inventory System to manage resources for Star Event Center

Instructor: TS. Nguyen Phuong Anh

Class: INS306101

Group: 1

Group member:

- Vu Dinh Manh - 20070747
- Pham Tuan Dung - 20070911
- Phung Linh Chi - 21070688
- Luong Thanh Hang - 20070709
- Pham Ngoc Anh - 22070566

Hanoi, 2024

Abstract

In the ever-evolving landscape of event organization, the quest for impeccable service provision amidst heightened living standards has become paramount. However, amidst a plethora of event companies, the challenge of delivering excellence is compounded by fierce competition, inflated prices, and wavering quality. Enter Star Event, a beacon of distinction in the industry, offering a plethora of services at competitive rates with bespoke designs for every occasion. Yet, navigating the intricacies of inventory management poses formidable challenges for Star Event, exacerbated by decentralized storage facilities and the presence of perishable items. This report endeavors to offer bespoke solutions to address these challenges, ushering in a new era of streamlined operations and enhanced efficiency for Star Event. By implementing a tailored inventory management system, Star Event seeks to optimize its organizational structure, centralize management operations, and fortify its position as a paragon of reliability and excellence in event management. Through a concerted effort to align with the company's current needs, these solutions aim to alleviate bottlenecks, minimize risks, and foster sustainable growth for Star Event in the dynamic landscape of event organization.

TABLE OF CONTENTS

I. INTRODUCE

In today's rapidly developing society, people's living standards are increasingly high, making event organization one of the top concerns. With a vast number of event companies, the competitive market is large, but high prices and uncertain quality introduce many unpredictable risks. Star Event was established to become the most prestigious partner for customers. "Star Event - Create a moment, make a mark in life." Star Event provides a variety of services at reasonable prices with beautiful designs suitable for every event. Every stage of production is closely monitored by our company.

Managing various types of event equipment and supplies with their unique sizes, styles, and characteristics makes it difficult for Star Event to classify, organize, and track these items. Additionally, goods are stored in multiple warehouses without a centralized management system, making inspection and access challenging. Some items have short expiration dates, requiring strict monitoring to avoid expiration or damage. The lack of a systematic management system leads to incomplete and inaccurate inventory data, causing difficulties in monitoring, controlling, and retrieving information. Therefore, Star Event needs an information system to manage inventory effectively, minimizing the risk of loss and avoiding impacts on business operations.

This report aims to provide appropriate solutions to address the inventory management issue. We have decided to switch to a new inventory management system to meet the company’s current needs and resolve the ongoing backlog problem.

II. TARGET

The goal of this report is to propose an optimal inventory management system tailored to meet the specific needs of Star Event, a company specializing in event organization services. By implementing this new system, the company aims to enhance the classification and organization of its inventory more efficiently. The new system will develop the capability to accurately categorize and organize a wide range of event equipment and supplies, taking into account their sizes, styles, and unique characteristics. This will help Star Event manage its inventory scientifically, making it easier to locate and check equipment when needed, from audio and lighting to decorations and other event accessories.

Furthermore, centralized inventory management will simplify the storage and retrieval of equipment from multiple warehouses. This will improve access and inspection processes, reducing the time and effort required to manage dispersed warehouses. A centralized system will help unify information, prevent data discrepancies, and ensure high accuracy in inventory management.

The new system will also implement robust tracking mechanisms to monitor supplies with short expiration dates, such as fresh flowers, food, or batteries and chemicals. This will ensure that these supplies are used or replaced in a timely manner to avoid losses due to expiration or damage. Monitoring expiration dates is crucial to minimize waste and maintain service quality, providing peace of mind to customers about the quality of services they receive.

To ensure complete and accurate inventory data, all data will be recorded in real-time. This will facilitate precise and timely monitoring, control, and retrieval of information. A real-time data recording system will enhance inventory management efficiency, promptly identifying arising issues and enabling appropriate corrective actions.

The system will also help minimize the risk of equipment loss through improved tracking and management. This will prevent disruptions to business operations and reduce financial losses. Improved inventory management will

contribute to the overall business efficiency, helping Star Event maintain and sustainably grow.

Finally, a flexible and scalable inventory management solution will support Star Event's evolving business needs and growth. A flexible system will easily adapt to changes in the scale and scope of the company's operations, ensuring long-term operational efficiency. By addressing these objectives, Star Event aims to enhance its inventory management processes, minimize risks, and ultimately improve service quality, maintaining its reputation as a prestigious and reliable event organization company.

III. MODELLING

1. REA Process

+ Dual relationships in inventory purchases

When an event organization needs to import more goods from suppliers, two main events happen:

- Purchasing Inventory (Inflow):

- **Resources**: Event decorations like flowers, background, etc… and equipment like speakers, microphones, lighting, and so on. (increasing)
- **Event**: Receiving inventory from the supplier.
- **Agent**: Supplier (external), Purchasing Department (internal).

- Disbursing Payment (Outflow):

- **Resources**: Cash (decreasing)
- **Event**: Payment for the inventory to the supplier
- **Agent**: Supplier (external), Accounting Supervisor (internal)

+ Step-by-step Scenario

Star Event Company is a service company specializing in organizing events. We regularly purchase a variety of inventory such as decorations, lighting equipment, and catering supplies to prepare events according to customer requirements.

- Purchase inventory: Star Event Company needs additional decorations and lighting equipment from a supplier to prepare for a wedding held at sea.

Detailed steps:

- Order: The Purchasing Department orders 70 sets of decorations and 30 sets of lighting equipment.
- Receiving goods: The supplier delivers goods to the company's warehouse.
- Increase inventory: Inventory staff verifies deliveries (product quantity and quality) and updates inventory records to reflect the addition of 70 sets of decorations and 30 sets of lighting equipment.

- Payment Disbursement: Star Events Company pays 10 million VND to the supplier for the inventory received.

Detailed steps:

- Invoice receipt: The Accounts Payable Department received an invoice worth 10 million VND from the supplier.
- Payment processing: The company decides to pay immediately, down to 10,000 VND in cash.
- Cash reduction: The cash account is debited and the payment is recorded in the accounting system.

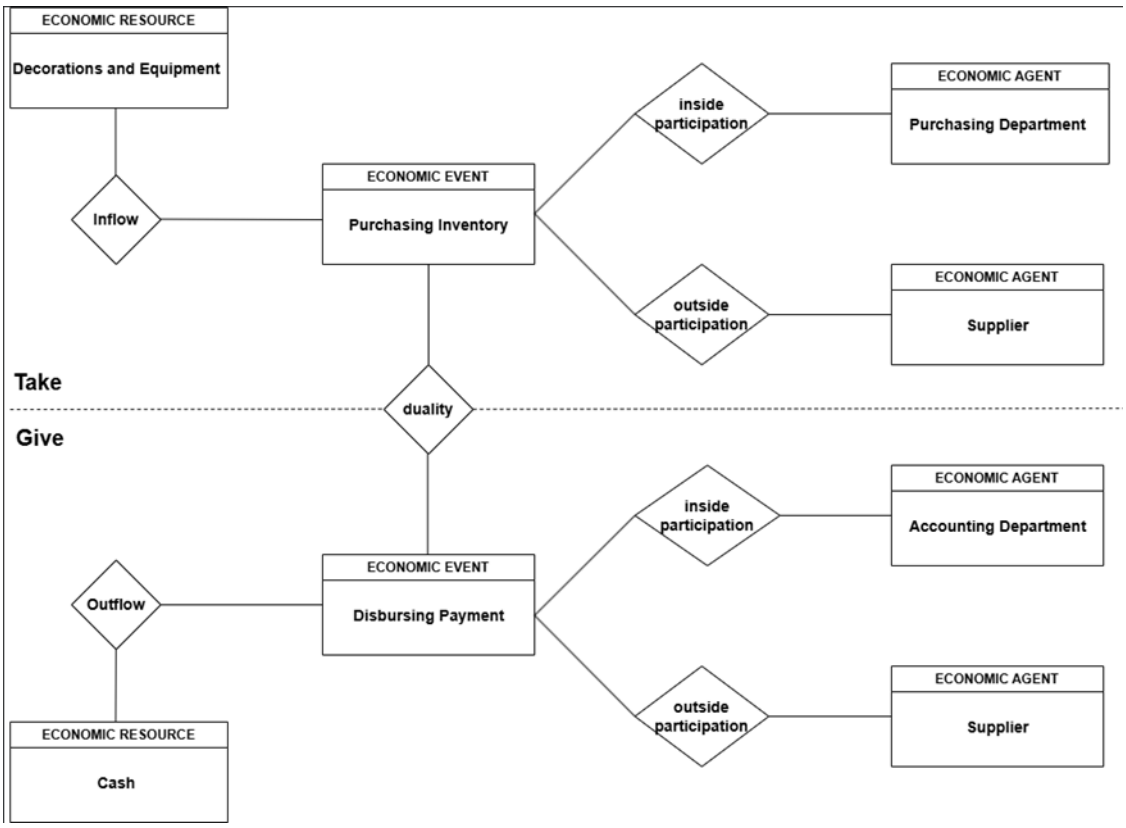


Figure 1. REA process

2. Business process

+ Our center overall business process is display as the following picture:

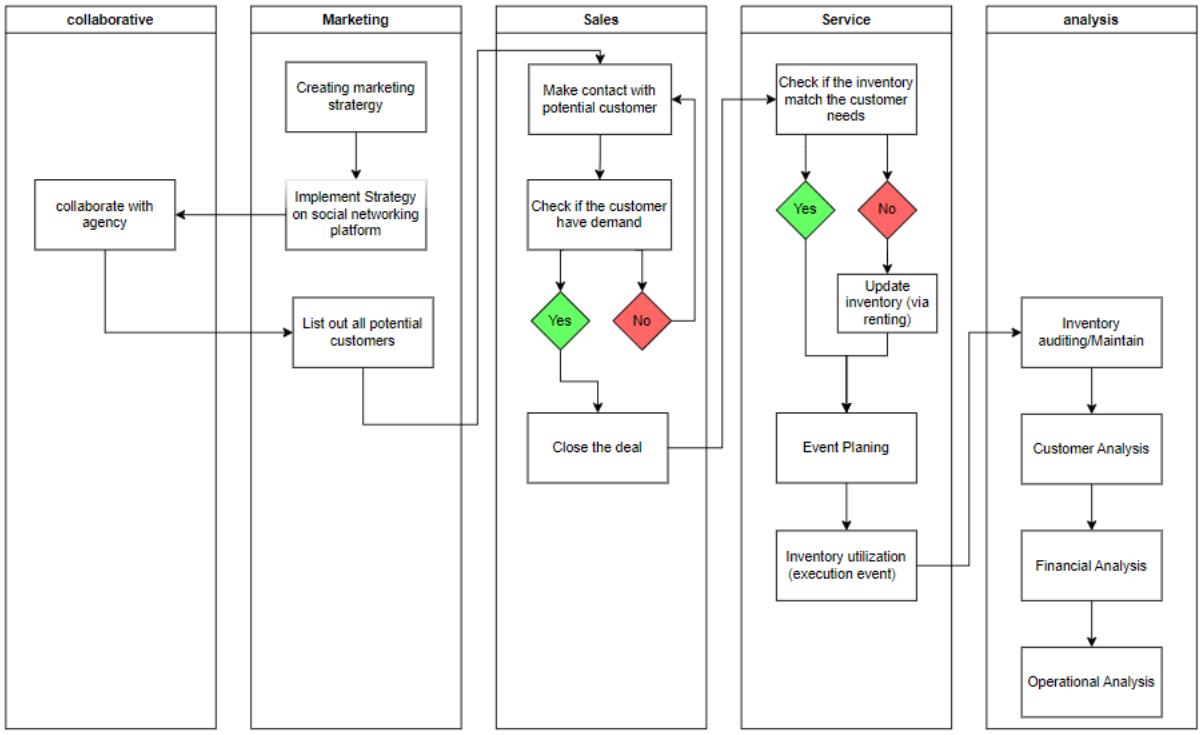


Figure 2. Business process

+ Here, we separate our business process into 5 different processes. They are:

- Collaborative:

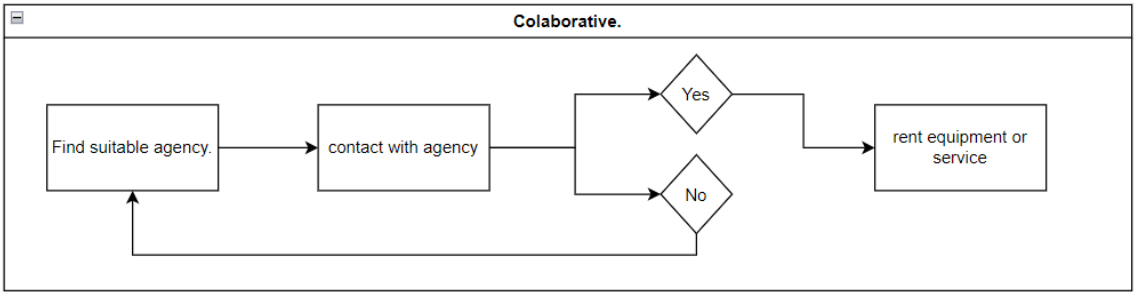


Figure 3. Collaborative

Through a collaborative process, we can leverage the expertise of various agencies. These agencies specialize in specific areas, such as marketing and inventory management. By working together, we can develop a comprehensive strategy to reach new customers, effectively update our stock items, and ultimately achieve our business goals.

- Marketing:

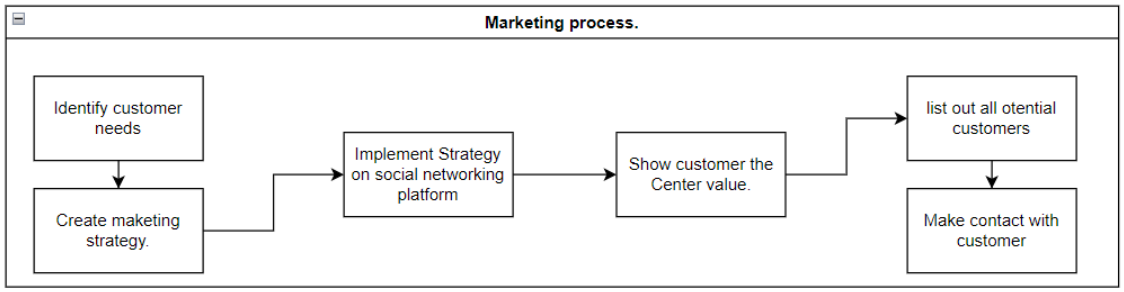


Figure 4. Marketing

The marketing process for events starts by delving into the minds of potential attendees. This is achieved through market research, surveys, focus groups, and social media listening. By gathering this data, we can paint a clear picture of:

- Event Goals: What are attendees hoping to achieve by attending? Is it professional development, networking opportunities, entertainment, or a combination?

- Target Audience: Who are the ideal attendees for this event? Understanding demographics, interests, and pain points allows for targeted marketing efforts.
- Desired Experience: What kind of atmosphere or experience are attendees seeking? This could range from a formal conference setting to a casual and interactive workshop.

Once we understand the "why" behind event attendance, we can craft a targeted marketing strategy. And by implementing the marketing strategy effectively, we can connect to potential customers.

- Sales:

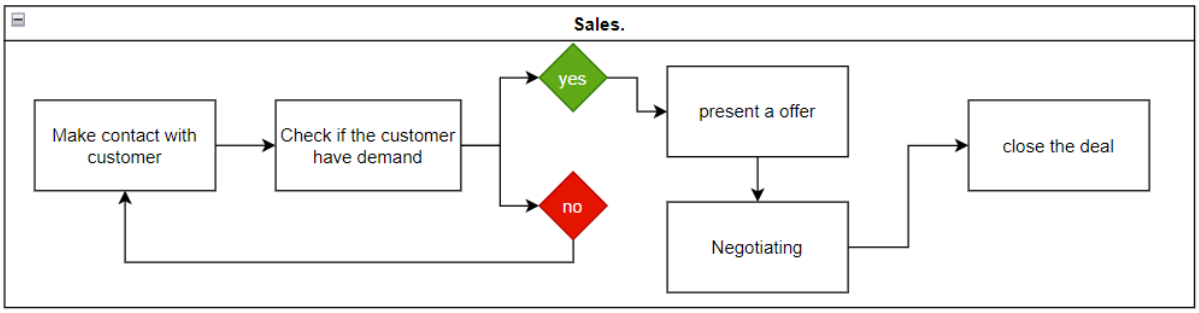


Figure 5. Sales

The sales process starts with making contact with potential customers. These potential customers, also known as prospects, could be facing challenges that our services are uniquely qualified to solve. During this initial contact, we aim to build rapport and understand their specific needs. By asking insightful questions, we can uncover their pain points and explore whether our services can provide a valuable solution.

If there's a good fit, we move on to the negotiation stage. This collaborative phase involves discussing the details of our offering, including the scope of services and pricing. It's a two-way street where we listen carefully to their questions and concerns, addressing them with clear explanations and demonstrations of the value we bring. The ultimate goal is to reach a mutually beneficial agreement that leaves both parties feeling confident and excited to

move forward. By successfully navigating these steps, we can turn those initial contacts into happy customers and fuel the growth of our business.

- Services:

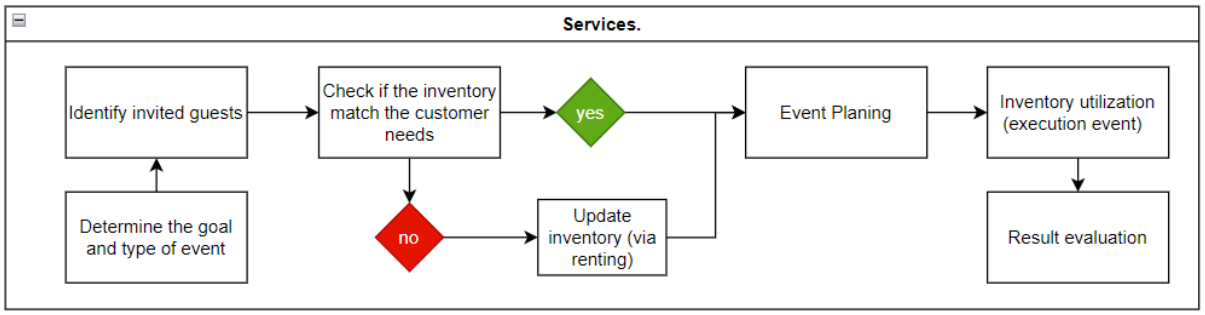


Figure 6. Services

In the service process, we prioritize meticulous planning to ensure a successful event. First, we take a detail-oriented approach, identifying the type of event and the guest list. Understanding the nature of the gathering, whether it's a formal business conference or a casual birthday party, allows us to tailor our approach. Knowing the guest count is equally important, as it helps us determine the scale of the event and the resources required.

Once we have a clear picture of the event, we spring into action, utilizing our well-stocked warehouse. Chairs, tables, tools, decorations – everything needed to bring the event to life is meticulously chosen from our inventory. This ensures a cohesive look and feel, while also guaranteeing the functionality and comfort needed for a smooth experience.

However, we understand that unforeseen needs can arise. If our inventory doesn't encompass every single item required, we don't hesitate to leverage our network of trusted collaborators. Through seamless rental arrangements, we can bridge any gaps and ensure all the necessary elements are present to make the event a resounding success. This collaborative approach allows us to cater to a wider range of events, offering maximum flexibility to our clients.

- Analysis:

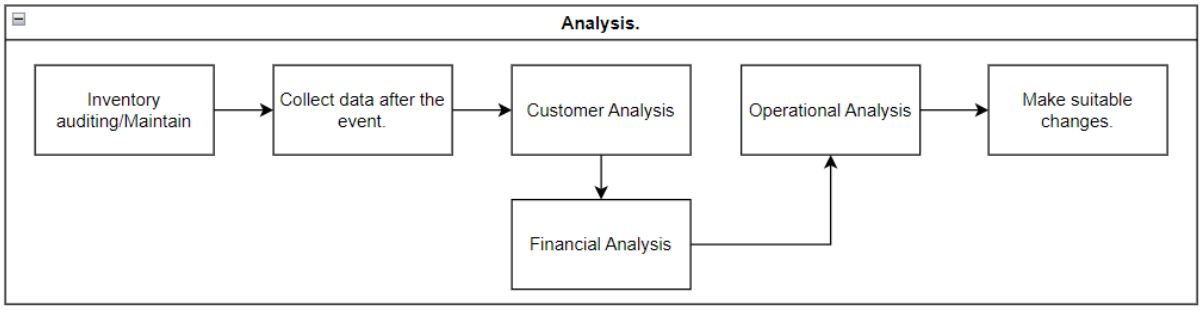


Figure 7. Analysis

The analysis process acts as the capstone to our workflow, providing invaluable insights after all the other stages are complete. It's like taking a step back and examining the big picture. By analyzing the data collected throughout the other processes, we can identify areas for improvement. This analysis helps us pinpoint inefficiencies, bottlenecks, or gaps in our current approach. With this knowledge, we can then strategically implement changes to optimize our processes and maximize their effectiveness.

3. System model

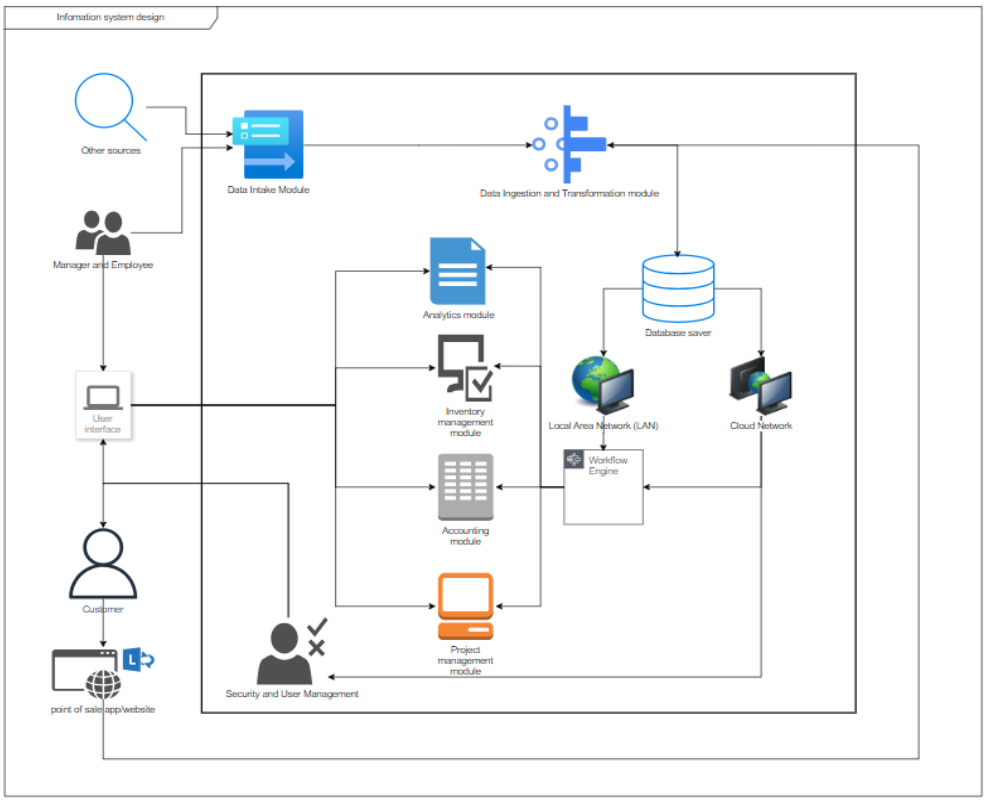


Figure 8. System model

This is the overall look of how our infomation system works. It starts by taking command from the users (Manager or employee of Star Event), the user then can create a new project(Event plan) or look into the inventory data. Our information system will take the data from the database and use it to interact with the user. Customers can also use the Interface to order an event and our information system can collect the needed data and store it in the database for the User to check.

IV. ER RELATIONS

1. ERD

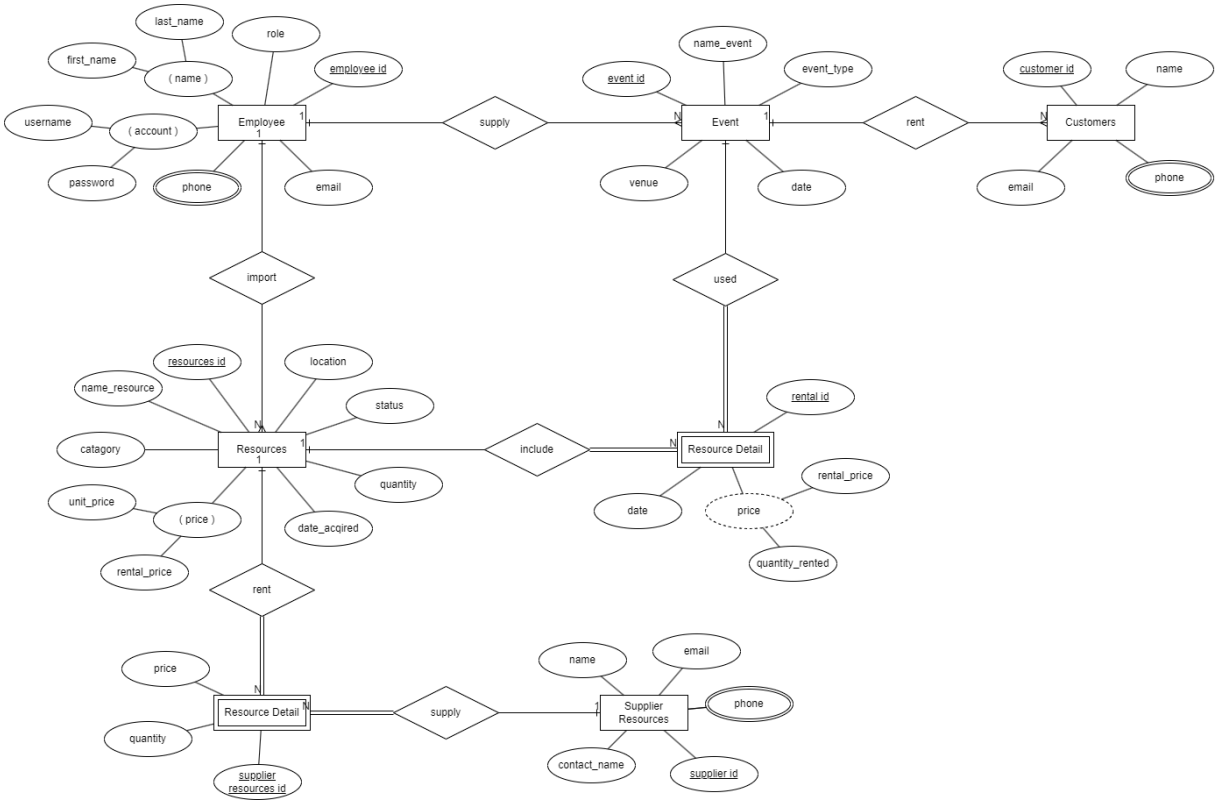


Figure 9. Entity relationship diagram

2. Database Diagram

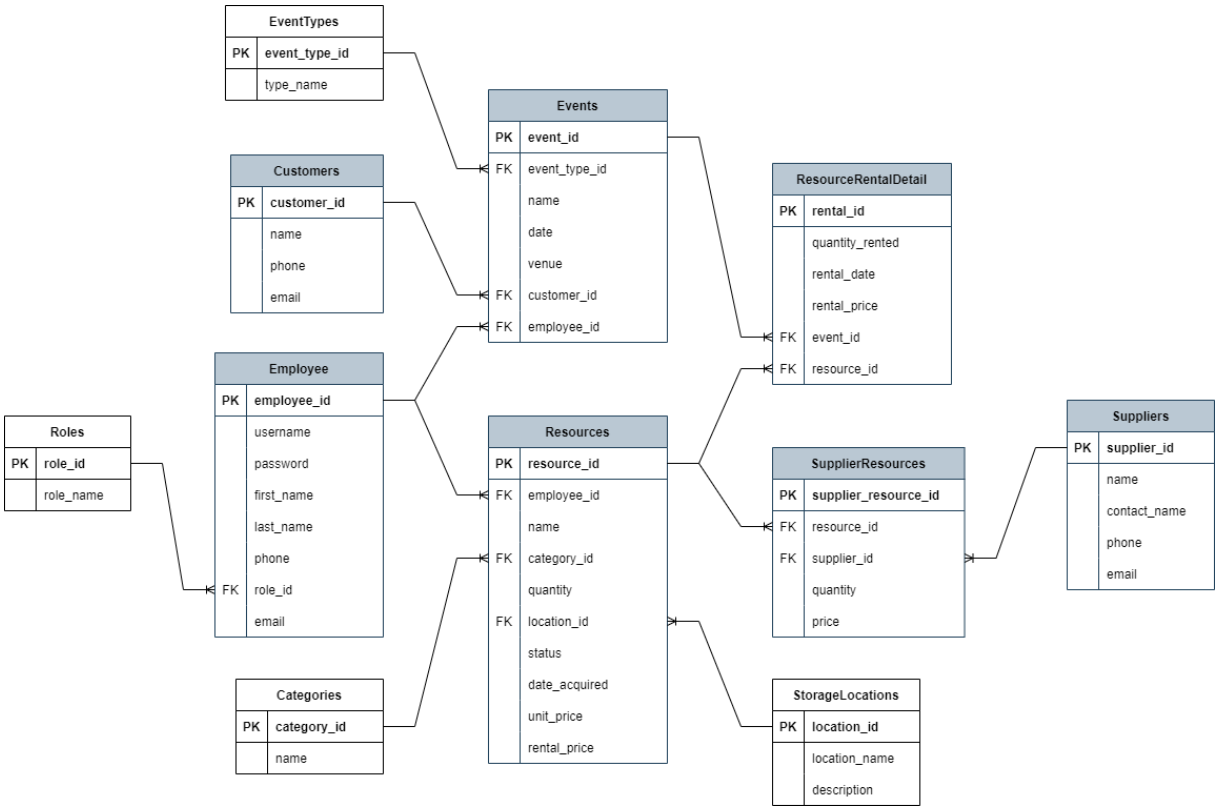


Figure 10. Database diagram

V. IMPLEMENTATIONS IN SQL SERVER

1. Database description

+ ***Table categories:*** this table stores information about different categories for resources.

- category_id (int, primary key): the category id, uniquely identifying each category.
- name (varchar(255), not null): the name of the category.

+ ***Table storagelocations:*** this table stores information about storage locations.

- location_id (int, primary key): the location id, uniquely identifying each location.

- location_name (varchar(255), not null): the name of the storage location.
- description (text): detailed description of the storage location.

+ ***Table customers:*** this table stores information about customers.

- customer_id (int, primary key): the customer id, uniquely identifying each customer.
- name (varchar(255), not null): the name of the customer.
- email (varchar(255), unique): the email address of the customer, must be unique.
- phone (varchar(20), not null, unique): the phone number of the customer, must be unique.

+ ***Table roles:*** this table stores information about employee roles.

- role_id (int, primary key): the role id, uniquely identifying each role.
- role_name (varchar(255), not null): the name of the role.

+ ***Table employees:*** this table stores information about employees.

- employee_id (int, primary key): the employee id, uniquely identifying each employee.
- username (varchar(50), not null, unique): the username of the employee, must be unique.
- password (varchar(255), not null): the password of the employee.
- first_name (varchar(255), not null): the first name of the employee.
- last_name (varchar(255), not null): the last name of the employee.
- phone (varchar(20), not null, unique): the phone number of the employee, must be unique.
- email (varchar(255), unique): the email address of the employee, must be unique.

- role_id (int, not null, foreign key): the role id of the employee, linked to the roles table.

+ ***Table resources:*** this table stores information about resources.

- resource_id (int, primary key): the resource id, uniquely identifying each resource.
- employee_id (int, foreign key): the id of the employee managing the resource, linked to the employees table.
- name (varchar(255), not null): the name of the resource.
- category_id (int, foreign key): the category id of the resource, linked to the categories table.
- quantity (int, not null): the quantity of the resource.
- location_id (int, foreign key): the location id of the resource, linked to the storagelocations table.
- status (varchar(50), not null): the status of the resource.
- date_acquired (datetime, not null): the date the resource was acquired.
- unit_price (decimal(10, 2), not null): the unit purchase price of the resource.
- rental_price (decimal(10, 2), not null): the rental price per unit of the resource.

+ ***Table event types:*** this table stores information about different types of events.

- event_type_id (int, primary key): the event type id, uniquely identifying each event type.
- type_name (varchar(255), not null): the name of the event type.

+ ***Table events:*** this table stores information about events.

- event_id (int, primary key): the event id, uniquely identifying each event.

- employee_id (int, foreign key): the id of the employee responsible for the event, linked to the employees table.
- customer_id (int, foreign key): the id of the customer for the event, linked to the customers table.
- event_type_id (int, foreign key): the event type id, linked to the eventtypes table.
- name (varchar(255), not null): the name of the event.
- date (datetime, not null): the date of the event.
- venue (varchar(255), not null): the venue of the event.

+ ***Table resourcerentaldetail:*** this table stores details about renting resources for events.

- rental_id (int, primary key): the rental id, uniquely identifying each rental.
- event_id (int, foreign key): the event id, linked to the events table.
- resource_id (int, foreign key): the id of the resource rented, linked to the resources table.
- quantity_rented (int, not null): the quantity of the resource rented.
- rental_date (datetime, not null): the date the resource was rented.
- rental_price (decimal(10, 2), not null): the rental price of the resource.

+ ***Table suppliers:*** this table stores information about suppliers.

- supplier_id (int, primary key): the supplier id, uniquely identifying each supplier.
- name (varchar(255), not null): the name of the supplier.
- contact_name (varchar(255)): the contact name of the supplier.
- phone (varchar(20), not null): the phone number of the supplier.
- email (varchar(255)): the email address of the supplier.

+ ***Table supplierresources:*** this table stores information about resources provided by suppliers.

- supplier_resource_id (int, primary key): the supplier resource id, uniquely identifying each supplier resource.
- supplier_id (int, foreign key): the supplier id, linked to the suppliers table.
- resource_id (int, foreign key): the resource id, linked to the resources table.
- quantity (int, not null): the quantity of the resource provided by the supplier.
- price (decimal(10, 2), not null): the price of the resource provided by the supplier.

2. SQL Code

+ ***Table categories:*** this table stores information about different categories for resources.

Create table	Insert data
CREATE TABLE Categories (category_id INT PRIMARY KEY, name VARCHAR(255) NOT NULL);	INSERT INTO Categories (category_id, name) VALUES (1, 'Sound Equipment'), (2, 'Lighting Equipment'), (3, 'Event Decorations'), (4, 'Catering Services');

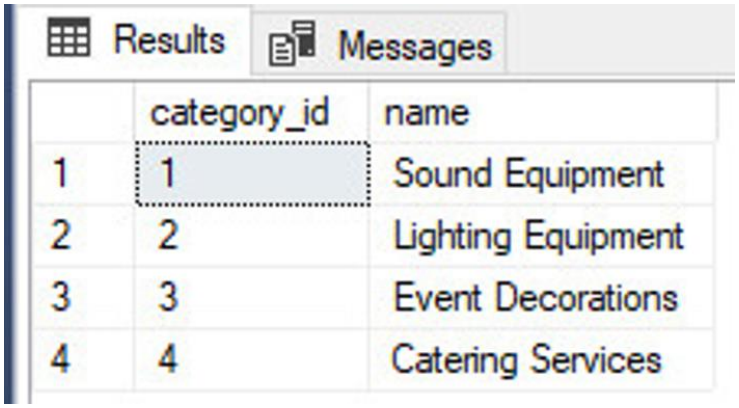
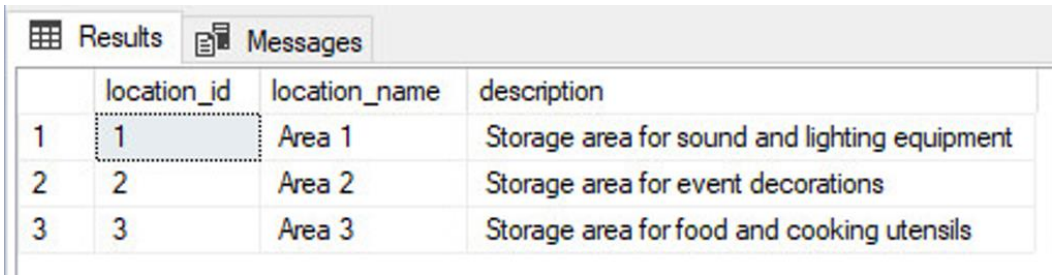


Figure 11. Table categories

+ ***Table storagelocations:*** this table stores information about storage locations.

Create table	Insert data
CREATE TABLE StorageLocations (location_id INT PRIMARY KEY, location_name VARCHAR(255) NOT NULL, description TEXT);	INSERT INTO StorageLocations (location_id, location_name, description) VALUES (1, 'Area 1', 'Storage area for sound and lighting equipment'), (2, 'Area 2', 'Storage area for event decorations'), (3, 'Area 3', 'Storage area for food and cooking utensils');



	location_id	location_name	description
1	1	Area 1	Storage area for sound and lighting equipment
2	2	Area 2	Storage area for event decorations
3	3	Area 3	Storage area for food and cooking utensils

Figure 12. Table storagelocations

+ ***Table customers:*** this table stores information about customers.

Create table	Insert data
CREATE TABLE Customers (customer_id INT PRIMARY KEY, name VARCHAR(255) NOT NULL, email VARCHAR(255) UNIQUE, phone VARCHAR(20) NOT NULL UNIQUE);	INSERT INTO Customers (customer_id, name, email, phone) VALUES (1, 'ABC Company', 'abc@example.com', '1234567890'), (2, 'XYZ Store', 'xyz@store.com', '0987654321'),

	(3, 'DEF Organization', 'def@organization.org', '1357924680');

	customer_id	name	email	phone
1	1	ABC Company	abc@example.com	1234567890
2	2	XYZ Store	xyz@store.com	0987654321
3	3	DEF Organization	def@organization.org	1357924680

Figure 13. Table customers

+ ***Table roles:*** this table stores information about employee roles.

Create table	Insert data
CREATE TABLE Roles (role_id INT PRIMARY KEY, role_name VARCHAR(255) NOT NULL);	INSERT INTO Roles (role_id, role_name) VALUES (1, 'Admin'), (2, 'Resource Manager'), (3, 'Inventory Clerk'), (4, 'Storage Staff'), (5, 'Event Staff'), (6, 'Logistics and Delivery Staff');

Results Messages

	role_id	role_name
1	1	Admin
2	2	Resource Manager
3	3	Inventory Clerk
4	4	Storage Staff
5	5	Event Staff
6	6	Logistics and Delivery Staff

Figure 14. Table roles

+ ***Table employees:*** this table stores information about employees.

Create table	Insert data
CREATE TABLE Employees (employee_id INT PRIMARY KEY, username VARCHAR(50) NOT NULL UNIQUE, password VARCHAR(255) NOT NULL, first_name VARCHAR(255) NOT NULL, last_name VARCHAR(255) NOT NULL, phone VARCHAR(20) NOT NULL, email VARCHAR(255), role_id INT NOT NULL, FOREIGN KEY (role_id) REFERENCES Roles(role_id));	INSERT INTO Employees (employee_id, username, password, first_name, last_name, phone, email, role_id) VALUES (1, 'admin', 'hashed_admin', 'Admin', 'User', '0123456789', NULL, 1), (20000001, '20000001', 'hashed01', 'John', 'Smith', '0123456789', 'john.smith@example.com', 2), (20000002, '20000002', 'hashed02', 'Emma', 'Johnson', '0123456789', 'emma.johnson@example.com', 4), (20000003, '20000003', 'hashed03', 'Michael', 'Brown', '0123456789', 'michael.brown@example.com', 5), (20000004, '20000004', 'hashed04', 'Sophia', 'Miller', '0123456789', 'sophia.miller@example.com', 6), (20000005, '20000005', 'hashed05', 'Daniel', 'Davis', '0123456789', 'daniel.davis@example.com', 3), (20000006, '20000006', 'hashed06', 'Olivia', 'Wilson', '0123456789', 'olivia.wilson@example.com', 4), (20000007, '20000007', 'hashed07', 'James', 'Taylor', '0123456789', 'james.taylor@example.com', 5),

	(20000008, '20000008', 'hashed08', 'Emily', 'Martinez', '0123456789', 'emily.martinez@example.com', 6), (20000009, '20000009', 'hashed09', 'William', 'Anderson', '0123456789', 'william.anderson@example.com', 3), (20000010, '20000010', 'hashed10', 'Isabella', 'Lopez', '0123456789', 'isabella.lopez@example.com', 4);

Results | Messages

	employee_id	username	password	first_name	last_name	phone	email	role_id
1	1	admin	hashed_admin	Admin	User	0123456789	NULL	1
2	20000001	20000001	hashed01	John	Smith	0123456789	john.smith@example.com	2
3	20000002	20000002	hashed02	Emma	Johnson	0123456789	emma.johnson@example.com	4
4	20000003	20000003	hashed03	Michael	Brown	0123456789	michael.brown@example.com	5
5	20000004	20000004	hashed04	Sophia	Miller	0123456789	sophia.miller@example.com	6
6	20000005	20000005	hashed05	Daniel	Davis	0123456789	daniel.davis@example.com	3
7	20000006	20000006	hashed06	Olivia	Wilson	0123456789	olivia.wilson@example.com	4
8	20000007	20000007	hashed07	James	Taylor	0123456789	james.taylor@example.com	5
9	20000008	20000008	hashed08	Emily	Martinez	0123456789	emily.martinez@example.com	6
10	20000009	20000009	hashed09	William	Anderson	0123456789	william.anderson@example.com	3
11	20000010	20000010	hashed10	Isabella	Lopez	0123456789	isabella.lopez@example.com	4

Figure 15. Table employees

+ ***Table resources:*** this table stores information about resources.

Create table	Insert data
CREATE TABLE Resources (resource_id INT PRIMARY KEY, employee_id INT, name VARCHAR(255) NOT NULL, category_id INT, quantity INT NOT NULL, location_id INT,	INSERT INTO Resources (resource_id, employee_id, name, category_id, quantity, location_id, status, date_acquired, unit_price, rental_price) VALUES (1, 20000001, 'Portable Speaker', 1, 10, 1, 'Available', '2023-01-15', 100.00, 20.00),

status VARCHAR(50) NOT NULL, date_acquired DATETIME NOT NULL, unit_price DECIMAL(10, 2) NOT NULL, rental_price DECIMAL(10, 2) NOT NULL, FOREIGN KEY (employee_id) REFERENCES Employees(employee_id), FOREIGN KEY (category_id) REFERENCES Categories(category_id), FOREIGN KEY (location_id) REFERENCES StorageLocations(location_id));	(2, 20000001, 'Moving Head Light', 2, 15, 1, 'Available', '2022-12-10', 500.00, 50.00), (3, 20000004, 'Backdrop', 3, 5, 2, 'In Use', '2023-02-20', 200.00, 100.00);

Results | Messages

	resource_id	employee_id	name	category_id	quantity	location_id	status	date_acquired	unit_price	rental_price
1	1	20000001	Portable Speaker	1	10	1	Available	2023-01-15 00:00:00.000	100.00	20.00
2	2	20000001	Moving Head Light	2	15	1	Available	2022-12-10 00:00:00.000	500.00	50.00
3	3	20000004	Backdrop	3	5	2	In Use	2023-02-20 00:00:00.000	200.00	100.00

Figure 16. Table resources

+ ***Table event types:*** this table stores information about different types of events.

Create table	Insert data
CREATE TABLE EventTypes (event_type_id INT PRIMARY KEY,	INSERT INTO EventTypes (event_type_id, type_name) VALUES

type_name VARCHAR(255) NOT NULL);	(1, 'Wedding'), (2, 'Birthday'), (3, 'Corporate Event'), (4, 'Concert');

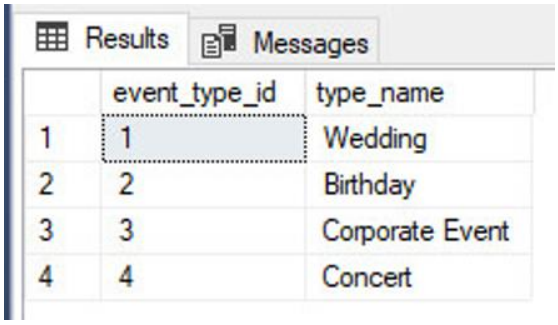


Figure 17. Table event types

+ ***Table events:*** this table stores information about events.

Create table	Insert data
CREATE TABLE Events (event_id INT PRIMARY KEY, employee_id INT, customer_id INT, event_type_id INT, name VARCHAR(255) NOT NULL, date DATETIME NOT NULL, venue VARCHAR(255) NOT NULL, FOREIGN KEY (employee_id) REFERENCES Employees(employee_id),	INSERT INTO Events (event_id, employee_id, customer_id, event_type_id, name, date, venue) VALUES (1, 20000007, 1, 3, 'Gala Dinner', '2024-05-15 18:00:00', 'Hotel A'), (2, 20000008, 2, 4, 'Fashion Exhibition', '2024-06-20 10:00:00', 'Exhibition Center B');

FOREIGN KEY (customer_id) REFERENCES Customers(customer_id), FOREIGN KEY (event_type_id) REFERENCES EventTypes(event_type_id));	

	event_id	employee_id	customer_id	event_type_id	name	date	venue
1	1	20000007	1	3	Gala Dinner	2024-05-15 18:00:00.000	Hotel A
2	2	20000008	2	4	Fashion Exhibition	2024-06-20 10:00:00.000	Exhibition Center B

Figure 18. Table events

+ ***Table resource rental detail:*** this table stores details about renting resources for events.

Create table	Insert data
CREATE TABLE ResourceRentalDetail (rental_id INT PRIMARY KEY, event_id INT NOT NULL, resource_id INT NOT NULL, quantity_rented INT NOT NULL, rental_date DATETIME NOT NULL, rental_price DECIMAL(10, 2) NOT NULL, FOREIGN KEY (event_id) REFERENCES Events(event_id),	INSERT INTO ResourceRentalDetail (rental_id, event_id, resource_id, quantity_rented, rental_date, rental_price) VALUES (1, 1, 1, 2, '2024-05-15 17:00:00', 40.00), (2, 1, 2, 4, '2024-05-15 16:00:00', 200.00), (3, 2, 3, 1, '2024-06-20 09:00:00', 100.00);

FOREIGN KEY (resource_id) REFERENCES Resources(resource_id));	

Results Messages

	rental_id	event_id	resource_id	quantity_rented	rental_date	rental_price
1	1	1	1	2	2024-05-15 17:00:00.000	40.00
2	2	1	2	4	2024-05-15 16:00:00.000	200.00
3	3	2	3	1	2024-06-20 09:00:00.000	100.00

Figure 19. Table resource rental detail

+ ***Table suppliers:*** this table stores information about suppliers.

Create table	Insert data
CREATE TABLE Suppliers (supplier_id INT PRIMARY KEY, name VARCHAR(255) NOT NULL, contact_name VARCHAR(255), phone VARCHAR(20) NOT NULL, email VARCHAR(255));	INSERT INTO Suppliers (supplier_id, name, contact_name, phone, email) VALUES (1, 'Supplier A', 'Alice Johnson', '2345678901', 'alice.johnson@supplierA.com'), (2, 'Supplier B', 'Bob Williams', '3456789012', 'bob.williams@supplierB.com');

Results Messages

	supplier_id	name	contact_name	phone	email
1	1	Supplier A	Alice Johnson	2345678901	alice.johnson@supplierA.com
2	2	Supplier B	Bob Williams	3456789012	bob.williams@supplierB.com

Figure 20. Table suppliers

+ ***Table supplierresources:*** this table stores information about resources provided by suppliers.

Create table	Insert data
CREATE TABLE SupplierResources (supplier_resource_id INT PRIMARY KEY, supplier_id INT, resource_id INT, quantity INT NOT NULL, price DECIMAL(10, 2) NOT NULL, FOREIGN KEY (supplier_id) REFERENCES Suppliers(supplier_id), FOREIGN KEY (resource_id) REFERENCES Resources(resource_id));	INSERT INTO SupplierResources (supplier_resource_id, supplier_id, resource_id, quantity, price) VALUES (1, 1, 1, 20, 18.00), (2, 2, 2, 10, 45.00), (3, 1, 3, 8, 90.00);

Results Messages

	supplier_resource_id	supplier_id	resource_id	quantity	price
1	1	1	1	20	18.00
2	2	2	2	10	45.00
3	3	1	3	8	90.00

Figure 21. Table supplierresources

3. Sql question

+ List all employees with the role of "Resource Manager"

```
SELECT *
FROM Employees
WHERE role_id = (SELECT role_id FROM Roles WHERE role_name = 'Resource Manager');
```

Results Messages

	employee_id	username	password	first_name	last_name	phone	email	role_id
1	20000001	20000001	hashed01	John	Smith	0123456789	john.smith@example.com	2

Figure 22

+ Find all resources with inventory less than 10

```
SELECT *
FROM Resources
WHERE quantity < 10;
```

Results Messages

	resource_id	employee_id	name	category_id	quantity	location_id	status	date_acquired	unit_price	rental_price
1	3	20000004	Backdrop	3	5	2	In Use	2023-02-20 00:00:00.000	200.00	100.00

Figure 23

+ Displays names, emails and phone numbers of all customers

```
SELECT name, email, phone
FROM Customers;
```

Results Messages

	name	email	phone
1	ABC Company	abc@example.com	1234567890
2	XYZ Store	xyz@store.com	0987654321
3	DEF Organization	def@organization.org	1357924680

Figure 24

+ Find the names and prices of all resources provided by "Supplier A"

```
SELECT Resources.name, SupplierResources.price
FROM Resources
INNER JOIN SupplierResources ON Resources.resource_id =
SupplierResources.resource_id
INNER JOIN Suppliers ON SupplierResources.supplier_id =
Suppliers.supplier_id
WHERE Suppliers.name = 'Supplier A';
```

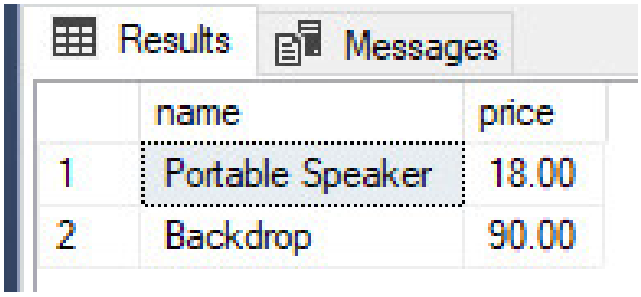


Figure 25

+ Lists the name, event type, and location of all events

```
SELECT Events.name, EventTypes.type_name, Events.venue
FROM Events
INNER JOIN EventTypes ON Events.event_type_id =
EventTypes.event_type_id;
```

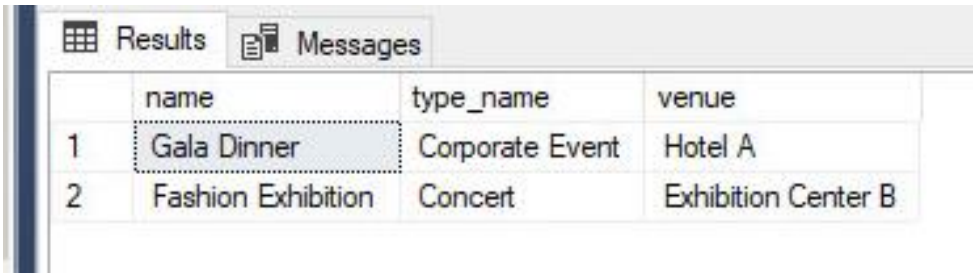


Figure 26

+ List the names and rental quantities of all resources rented for the event whose event_id is 1

```
event_id là 1
SELECT Resources.name, ResourceRentalDetail.quantity_rented
FROM ResourceRentalDetail
INNER JOIN Resources ON ResourceRentalDetail.resource_id =
Resources.resource_id
WHERE event_id = 1;
```

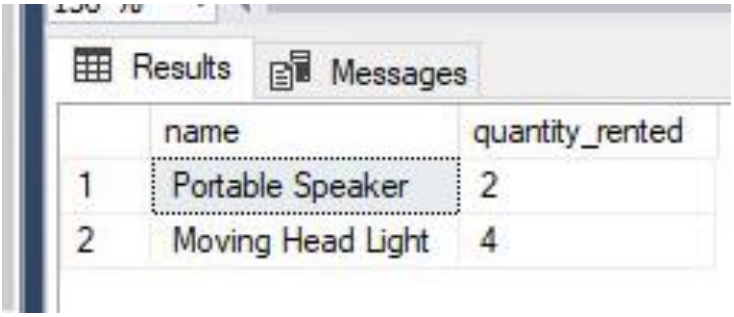


Figure 27

+ Add new resources to the Resources table

```
INSERT INTO Resources (resource_id, employee_id, name, category_id,
quantity, location_id, status, date_acquired, unit_price, rental_price) VALUES
(4, 20000002, 'Wireless Microphone', 1, 20, 1, 'Available', '2024-06-01', 150.00,
25.00),
(5, 20000003, 'LED Uplight', 2, 30, 1, 'Available', '2024-05-20', 75.00, 15.00);
```

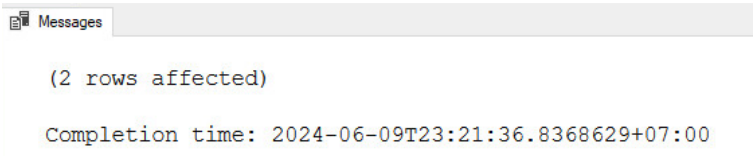


Figure 28

+ Add a new event to the Events table

```
INSERT INTO Events (event_id, employee_id, customer_id, event_type_id, name, date, venue) VALUES
(3, 20000005, 3, 1, 'Wedding Reception', '2024-07-10 17:00:00', 'Banquet Hall C'),
(4, 20000006, 1, 2, 'Corporate Meeting', '2024-08-01 09:00:00', 'Conference Room D');
```

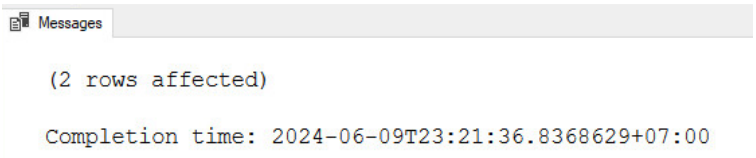


Figure 29

VI. USER INTERFACE

+ Log in:

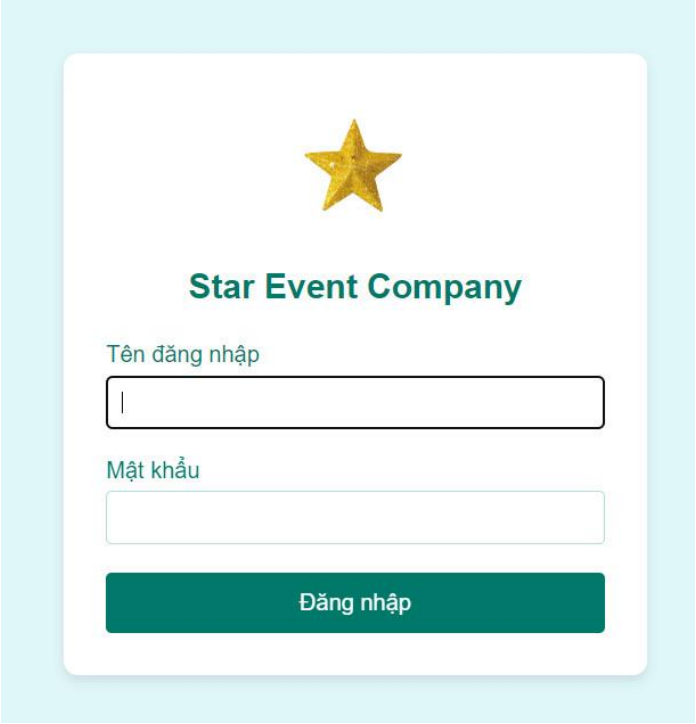


Figure 30. Log in

+ User interface:

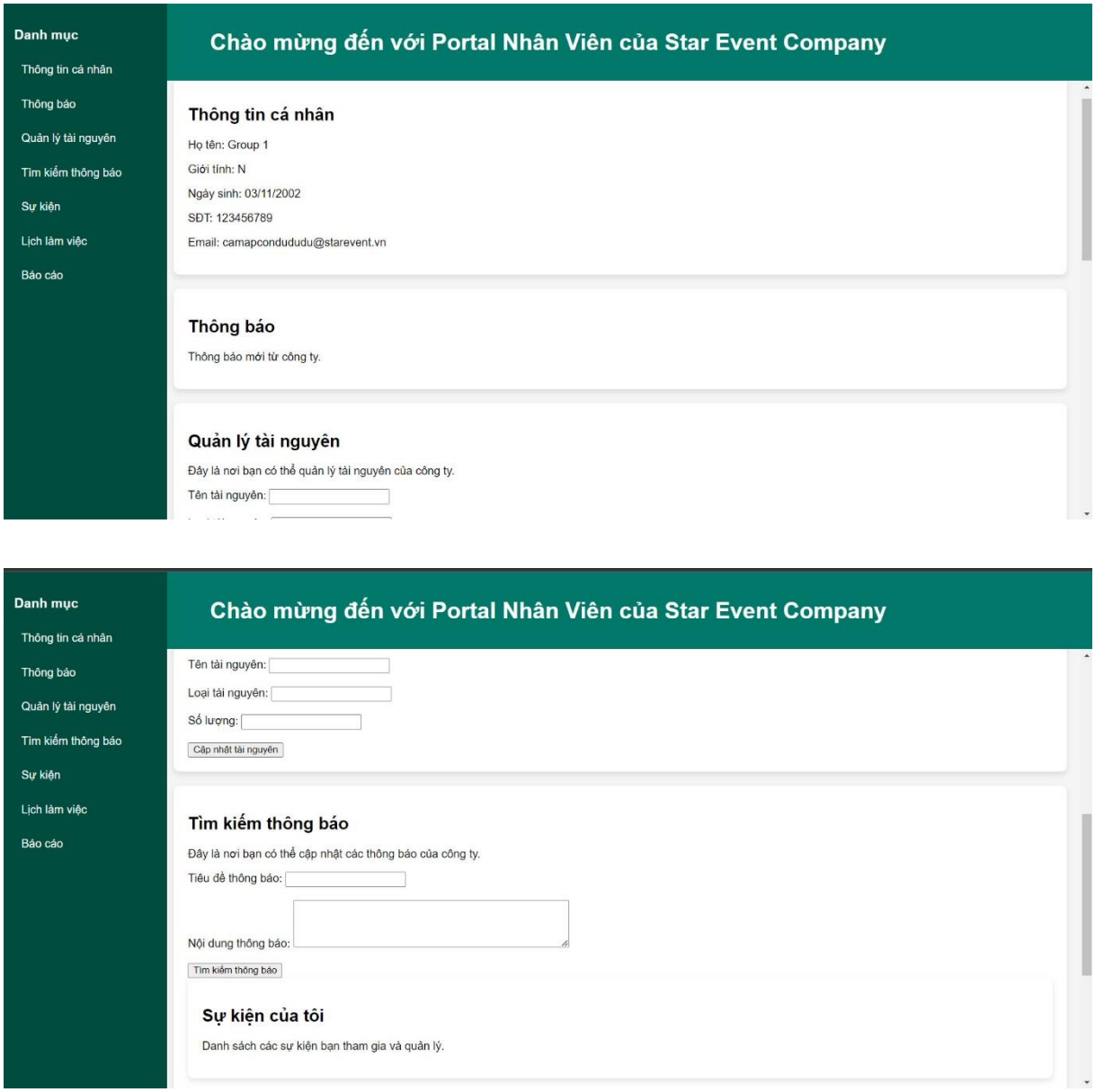


Figure 31. Personal information

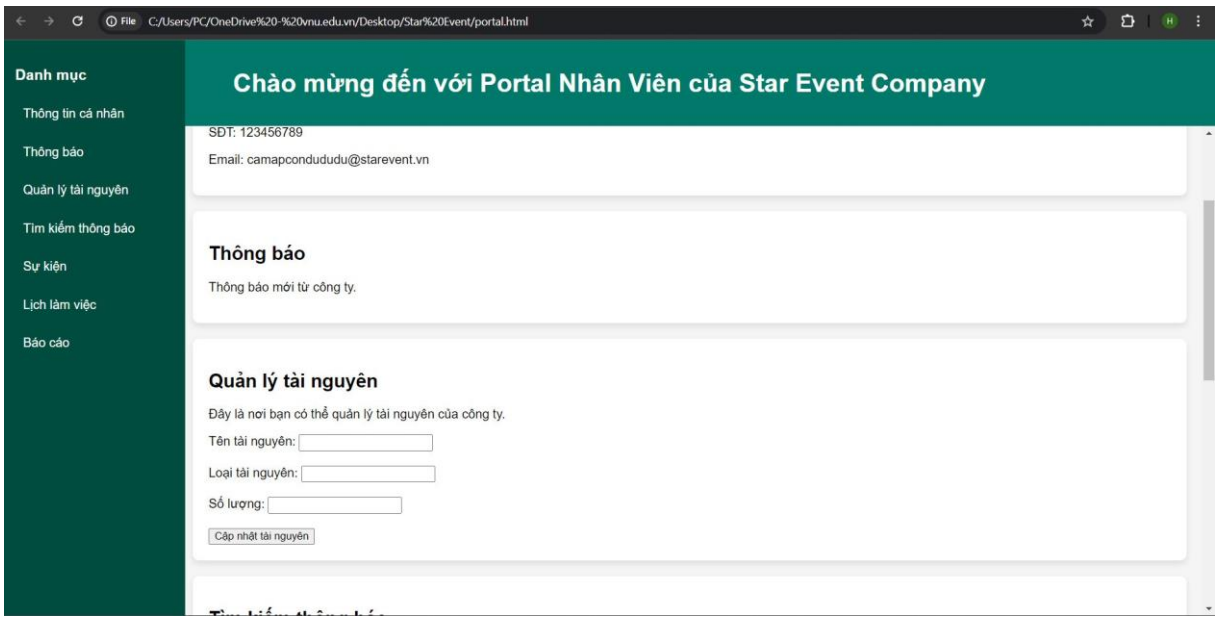


Figure 32. Resource management

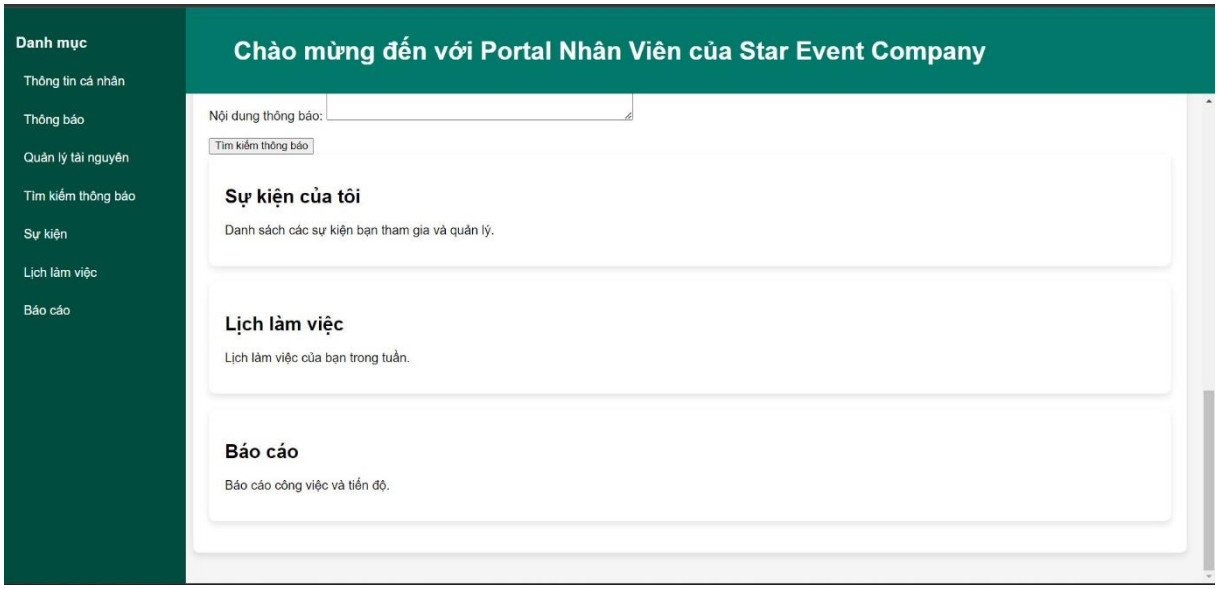


Figure 33. Event management

VII. CONCLUDE

Based on the proposed system implementation solutions above, it is undeniable that inventory management plays a crucial role for Star Event and the event organization industry as a whole. In an increasingly competitive landscape and with growing expectations from customers, ensuring service quality and operational flexibility has become paramount.

Inventory management involves more than just controlling stock levels; it requires an effective system for organization, monitoring, and management of information. Particularly in the event organization sector, where each event may require unique items and equipment, inventory management becomes more complex than ever.

By implementing the proposed solutions outlined in this report, Star Event will have the opportunity to optimize not only its own operations but also to make a significant difference in the industry. Concentrating on management, organizing warehouse systems, and applying rigorous monitoring measures for perishable items will create an efficient, flexible, and reliable working environment. This will not only help Star Event minimize losses but also enhance its ability to respond quickly and flexibly to customer demands.

In conclusion, improving inventory management is not only a current priority but also a crucial investment for the future of Star Event. By maintaining a commitment to applying leading management standards and continuously improving, Star Event will continue to assert its position as one of the leading companies in the event organization field, delivering unforgettable experiences for its customers.

VIII. REFERENCES

[1]. (PDF) Resource-Event-Agent REA Modelling in Revenue Information System RiS Development: Smart Application for Direct-Selling Dealers and SMEs (researchgate.net)

[2]. (PDF) The REA Modeling Approach to Teaching Accounting Information Systems (researchgate.net)

[3]. Reference guide to build inventory management and forecasting solutions on AWS | AWS Big Data Blog (amazon.com)

[4]. Star Events - Sự kiện Ngôi Sao Việt - Hololab

IX. CONTRIBUTION

Member	Task	Percent
Vu Dinh Manh	ER relations, SQL	20%
Pham Tuan Dung	Modelling	20%
Phung Linh Chi	Modelling	20%
Luong Thanh Hang	User interface	20%
Pham Ngoc Anh	SQL	20%
Total		100%