Republic of the Philippines

**CARMEN MUNICIPAL COLLEGE**

Poblacion Norte, Carmen, Bohol

**HAVEN BREW CAFÉ**

**GRACE P. BERONDO**

**JANICE DAG-UM**

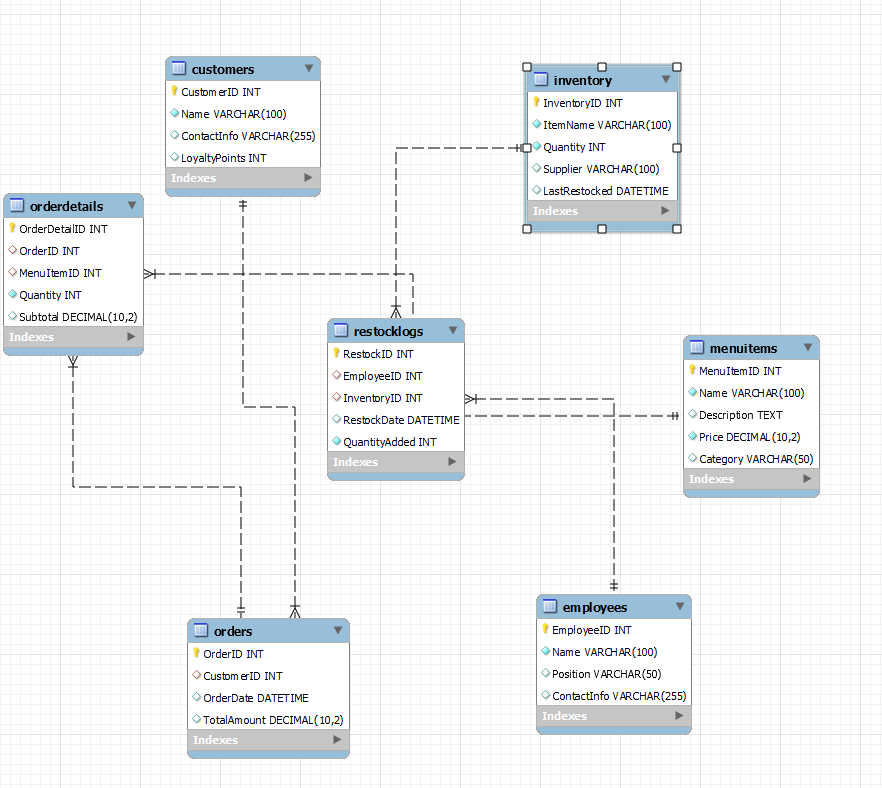
In Partial Fulfillment of the subject requirements

IS-104 System Analysis and Design

1. **Introduction**

The Haven Brew Café Database system is designed to efficiently manage customer orders, employee operations, menu items, and inventory tracking. This database aims to streamline the ordering process, track sales, monitor inventory levels, and facilitate employee management.

1. **Entity Relationship Diagram**

****

1. **DATABASE**

**Table Name: Customers**

|  |  |  |
| --- | --- | --- |
| FIELD NAME | TYPE (SIZE) | DESCRIPTION |
| CustomerID | INT  (AUTO\_INCREMENT) | Unique identifier of the customer (Primary Key). |
| NAME | VARCHAR(50) | Name of the customer. |
| CONTACT\_INFOR |  |  |
| LOYALTY\_POINT | INT | Loyalty points earned by the customer. |

**TABLE NAME: EMPLOYEES**

|  |  |  |
| --- | --- | --- |
| FIELD NAME | TYPE (SIZE) | DESCRIPTION |
| EmployeeID | INT  (AUTO\_INCREMENT) | Unique identifier of the employee (Primary Key). |
| Name | VARCHAR(50) | Name of the employee. |
| Position | VARCHAR(55) | Position or role of the employee. |
| Contact\_Info | VARCHAR(50) | Contact details of the employee. |

**TABLE NAME: ORDERS**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| OrderID | INT  (AUTO\_INCREMENT) | Unique identifier for each order (Primary Key). |
| CustomerID | INT | The customer who placed the order (Foreign Key). |
| Order\_date | DATETIME | Date and time when the order was placed. |
| Total\_amount | DECIMAL | Total amount of the order. |

**TABLE NAME: ODER DETAILS**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Order\_detailID | INT  (AUTO\_INCREMENT) | Unique identifier for each order detail (Primary Key). |
| OrderId | INT | Order to which this detail belongs (Foreign Key). |
| Menu\_itemID | INT | The menu item ordered (Foreign Key). |
| Quantity | INT | Number of items ordered. |
| Subtotal | DECIMAL | Total cost for this menu item. |

**TABLE NAME: MENU ITEMS**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Menu\_itemID | INT  (AUTO\_INCREMENT) | Unique identifier of the menu item (Primary Key). |
| Name | VARCHAR(50) | Name of the menu item. |
| Description | TEXT | Description of the menu item. |
| Price | DECIMAL | Price of the menu item. |
| Category | VARCHAR(100) | Category of the menu item (e.g., Beverage, Snack). |

**TABLE NAME: INVENTORY**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| InventoryID | INT  (AUTO\_INCREMENT) | Unique identifier of the inventory item (Primary Key). |
| Item\_name | VARCHAR(100) | Name of the inventory item. |
| Quantity | INT | Quantity available in stock. |
| Supplier | VARCHAR(100) | Supplier of the inventory item. |
| Last\_restocked | DATETIME | Date when the item was last restocked. |

**TABLE NAME: RESTOCK\_LOGS**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| RestockID | INT  (AUTO\_INCREMENT) | Unique identifier of the restock entry (Primary Key). |
| EmloyeeID | INT | Employee who performed the restock (Foreign Key). |
| InventoryID | INT | Inventory item that was restocked (Foreign Key). |
| Restock\_date | DATETIME | Date and time of restocking. |
| Quantity\_added | INT | Number of units added to inventory. |

1. **Functionalities**

The Brew Haven Cafe Database System provides several functionalities to streamline cafe operations:

1. Customer Management

Add, update, and delete customer records.

Track customer loyalty points for promotional offers.

2. Order Processing

Customers can place orders.

Employees can process and update order statuses.

Generate invoices for completed orders.

3. Inventory Management

Track stock levels of ingredients and supplies.

Employees can log restocking of items.

Automatic updates on the last restocked date when new stock is added.

4. Menu Management

Add, update, or remove menu items.

Categorize menu items (e.g., Coffee, Pastries).

Link menu items with inventory for stock monitoring.

5. Employee Management

Store employee details, including contact information and roles.

Track which employees process orders and restock inventory.

6. Reporting and Analytics

Generate sales reports based on orders and revenue.

View top-selling menu items.

Monitor inventory usage trends to avoid shortages.

**Creating table:** this code shows the creation of the tables.

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

ContactInfo VARCHAR(255),

LoyaltyPoints INT DEFAULT 0

) ENGINE = InnoDB;

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Position VARCHAR(50),

ContactInfo VARCHAR(255)

) ENGINE = InnoDB;

CREATE TABLE Orders (

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID INT,

OrderDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

TotalAmount DECIMAL(10,2),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE SET NULL

) ENGINE = InnoDB;

CREATE TABLE MenuItems (

MenuItemID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Description TEXT,

Price DECIMAL(10,2) NOT NULL,

Category VARCHAR(50)

) ENGINE = InnoDB;

CREATE TABLE OrderDetails (

OrderDetailID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

MenuItemID INT,

Quantity INT NOT NULL,

Subtotal DECIMAL(10,2),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

FOREIGN KEY (MenuItemID) REFERENCES MenuItems(MenuItemID) ON DELETE CASCAD

) ENGINE = InnoDB;

CREATE TABLE Inventory (

InventoryID INT PRIMARY KEY AUTO\_INCREMENT,

ItemName VARCHAR(100) NOT NULL,

Quantity INT NOT NULL,

Supplier VARCHAR(100),

LastRestocked DATETIME DEFAULT CURRENT\_TIMESTAM

) ENGINE = InnoDB;

CREATE TABLE RestockLogs (

RestockID INT PRIMARY KEY AUTO\_INCREMENT,

EmployeeID INT,

InventoryID INT,customerscustomers,

RestockDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

QuantityAdded INT NOT NULL,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID) ON DELETE SET NULL,

FOREIGN KEY (InventoryID) REFERENCES Inventory(InventoryID) ON DELETE CASCADE

) ENGINE = InnoDB;

**Inserting Users Information:** this code represents the user’s information input.

INSERT INTO Customers (Name, ContactInfo, LoyaltyPoints)

VALUES ('Alice Johnson', 'alice@email.com', 50),

('Bob Smith', 'bob@email.com', 20);

INSERT INTO Employees (Name, Position, ContactInfo)

VALUES ('John Doe', 'Barista', 'john@email.com'),

('Jane Doe', 'Manager', 'jane@email.com');

INSERT INTO MenuItems (Name, Description, Price, Category)

VALUES ('Espresso', 'Strong black coffee', 2.50, 'Beverage'),

('Cappuccino', 'Espresso with steamed milk foam', 3.50, 'Beverage'),

('Muffin', 'Blueberry muffin', 2.00, 'Food');

INSERT INTO Orders (CustomerID, TotalAmount)

VALUES (1, 6.00),

(2, 5.50);

INSERT INTO OrderDetails (OrderID, MenuItemID, Quantity, Subtotal)

VALUES (1, 1, 1, 2.50),

(1, 3, 1, 2.00),

(2, 2, 1, 3.50);

INSERT INTO Inventory (ItemName, Quantity, Supplier, LastRestocked)

VALUES ('Coffee Beans', 50, 'Local Supplier', '2025-03-10'),

('Milk', 30, 'Dairy Farm', '2025-03-12');

INSERT INTO RestockLogs (RestockID, EmployeeID, InventoryID, RestockDate, QuantityAdded)

Values (1, 1, 1, '2025-03-16 14:00:00', 20);

**Displaying:** this code represents the users filtered information.

SELECT \* FROM Customers;

SELECT \* FROM Employees;

SELECT \* FROM Orders;

SELECT \* FROM MenuItems;

SELECT \* FROM OrderDetails;

SELECT \* FROM Inventory;

SELECT \* FROM RestockLogs;