Information Systems and Data Modeling – IT1090

Assignment 2

Title: Online Dry-Cleaning and Laundry Services	
Batch Number: Y1S2/23/MTR/Gr02	Group Number: 02

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged. We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

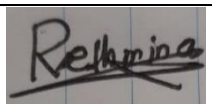
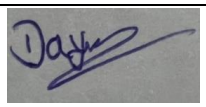
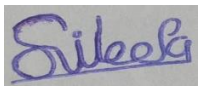
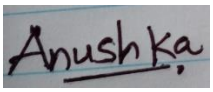
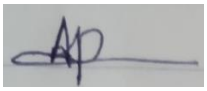
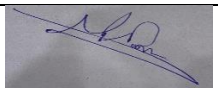
	Student Registration Number	Student Name	Signature
1	IT22253958	W.P.R. Nethmina	
2	IT22296078	Sarithmal K.D	
3	IT22226532	D.V.D Hashan	
4	IT22635952	Abeywickrama A.S.	
5	IT22244352	Hewahalpage	
6	IT22371522	G.H.P Iroshan	

Table of Contents

1. Introduction

"Our Dry-Cleaning System database is the backbone of our modern laundry and cleaning services. In a world increasingly driven by technology, this digital platform plays a pivotal role in streamlining our operations and delivering impeccable service to our valued customers.

At its core, our database is a virtual repository that securely stores and manages essential information. It acts as a central hub, enabling us to efficiently oversee and coordinate every aspect of our dry cleaning and laundry services.

Within this database, we meticulously record and organize customer data, ensuring that we have up-to-date contact information, preferences, and special instructions readily available. This empowers us to provide personalized and convenient services tailored to individual needs.

Furthermore, our database tracks every customer order from inception to completion. It records order details, item specifications, delivery preferences, and payment information. This meticulous documentation guarantees accuracy and timeliness in our service delivery.

Our employee management system is also an integral part of this database. It maintains comprehensive employee profiles, roles, and contact details, allowing us to optimize workforce deployment and ensure smooth daily operations across our outlets.

Financial transactions, including payments and invoices, are securely processed through the database. It integrates seamlessly with our banking partner, offering customers a hassle-free and secure payment experience.

In essence, our Dry-Cleaning System database is the cornerstone of our commitment to efficiency, convenience, and customer satisfaction. It empowers us to provide top-notch dry cleaning and laundry services while embracing the digital age.

2. Hypothetical Scenario.

Scenario: **Online Dry Cleaning and Laundry Services.**

In our online dry cleaning and laundry service system, we have various entities that interact with each other:

Customer: Customers can register with their Customer ID, name, phone number, email, date of birth, age. They can place orders for laundry and dry-cleaning services through an outlet as well as they can place online orders.

Manager: Each outlet has a manager responsible for overseeing operations, employees, payments, Marketing, supplies at that location. Manager has manger ID, name, phone number, date of birth, email, address.

Employees: Employees work in different outlets. Employees have Employee Id, name, phone number, email, Dob, age, address.

Outlet: Outlets are where the cleaning and laundry services take place. Each outlet has a unique ID, location, and contact details.

Dependent: Employees can have dependents linked to their accounts, such as family members sharing the same address. If an employee resigns, the company does not keep track of their dependents. Dependents have dependent ID, name, address, dob.

Supplier: Suppliers provide cleaning materials. They have Supplier ID, name, phone number, email.

Marketer: Marketers work on advertising and promotions. They have Marketer ID, name, phone, email, DOB.

Payment: Payments are made by customers for their orders. Each payment is associated with a customer and includes transaction details. Payment has Payment ID, type, amount, payment method, invoice number, date as attributes.

Item: Items include clothes, which are part of customer orders. Each item is linked to a specific order. Item has Item number, item name, item type, quantity, description.

Service: Services offered include "Laundry" and "Dry Cleaning," each with their pricing and delivery options. Services have Service ID, service name, description, price, duration.

3 Requirement Analysis.

3.1 Main Requirements.

3.1.1 Functional Requirements.

Main function of the system and events that take place between the users and the system is described by the Functional Requirements. Seven Users are using Online dry Cleaning and Laundry Services system. Namely: Guest, Registered User, Administrator, Manager, Bank. They Access this system in different ways where it is related to them.

1. **Guest and Registered User (They can access front-end of the system).**

User Requirements

- The services and other details of the system can be checked without accessing the system.
- Guest users can register to the system by providing required details for the registration.
- Guest views the available feedback.
- The outlet can be checked as per the requirement of a guest.
- The guest can check the rules and regulations in the system.
- Registered user login to the system by providing required user login credentials.
- Registered users can update their user profile information.
- Registered users can check all the services available in the system.
- Registered users can access convenient payment methods to make payment for required services.
- Registered users can check their order details and order history.

System Requirements

- The system should have easy navigation and a good user interface for unregistered users.
- For unregistered users the services available in the system should be shown.
- For unregistered users the system should display clear information about available packages.
- The system requires you to approve the registration details and create a user account.
- The system should allow unregistered users to register to the system.
- The username and password provided by the registered customer must be validated in the system.

- The system shall update and show registered users when their user profile information is updated.
- The system should demonstrate convenient payment methods for registered users to make payments for the services they require.
- All services available in the system should be displayed and registered users should be able to access the services.
- The system shall allow the registered user to change the services or cancel the services, when necessary, after placing an order.

2. **Administrator (can access the back end of the system).**

User Requirements

- Administrator signs into the website by providing required login credentials.
- Administrator can approve the reservations.
- Administrator can add and remove staff accounts.
- Administrator can activate and deactivate user accounts.
- Administrator can update drycleaning and laundry details.
- Administrator checks member feedbacks, reviews, and contacts.

System Requirements

- System should validate the user login credentials.
- System should delete details of the deleted by the user.
- System should update the details of the downgraded users and modified accounts in the database.
- System stores the member feedback, reviews and contacts and display them.
- The system should validate login credentials entered by the administrator.
- The system should generate reports for the administrator.

3. **Manager (can access the back end of the system).**

User Requirements

- Accept customer payments and manage refunds.
- Manager financial transactions can be initiated and monitored.
- The manager can assign orders to specific employees or departments.
- The manager can monitor all the orders and arrange for them to be fulfilled.
- A manager can assign specific roles and responsibilities to employees.
- Manager can manage salaries of employees.
- The manager can get notifications for low stock inventory items.

- The manager can process reorder orders and manage inventory replenishments.
- Manager can access customer payment information.
- Manager can generate reports or statements related to customer payments.

System Requirements

- The manager should implement a secure login system with unique passwords.
- Acceptance of customer payments or refunds must be enabled in the system.
- The system must ensure compliance with the necessary security protocols for secure data transmission and storage.
- A user-friendly interface should be set up in the system for the manager customer orders.
- The status of the orders should be updated in the system.
- The system should allow employees to place orders and track the progress of the orders.
- If the user removes or adds an inventory item, the system should remove or add it.
- Customer transaction processing shall be done by the system.
- System generation of customer payment related reports and statements.
- For completed customer payments, the manager should monitor the payments and the system should send an email or message to the customer to confirm the payment.

4. **Bank (can access payment processes).**

User Requirements

- Bank can ensure secure processing and storage of customer payment information and transactions.
- Bank can receive real-time updates on successful and failed payment transactions.
- bank can Implement fraud detection mechanisms to identify and prevent suspicious or unauthorized transactions.
- bank allow Integrate with payment gateways for seamless electronic fund transfers and card payments.
- bank allow facilitate easy reconciliation of funds with laundry service providers on a regular basis.

- Bank maintain detailed records of all laundry and dry cleaning-related financial transactions.
- The bank allows enable account creation, modification, and closure for laundry service providers' banking needs.
- Bank generates invoices or bills for laundry and dry-cleaning services on behalf of providers.
- Bank allow receive instant notifications for high-value transactions, chargebacks, and critical system events.
- bank allow access to payment history and transaction reports for auditing and analysis.

System Requirements

- The system must provide secure user authentication to ensure that only authorized bank personnel can access financial transactions and data.
- It should maintain a detailed transaction log, recording all financial interactions between the bank and the laundry system for auditing and dispute resolution.
- The system should support real-time integration with the bank's core banking system to ensure accurate balance updates and fund transfers.
- It must integrate a secure payment gateway for processing transactions, supporting multiple payment methods such as credit cards, wire transfers, and ACH.
- Define and enforce transaction limits to prevent unauthorized or excessive financial transfers between the bank and laundry service.
- Implement robust security measures to safeguard sensitive financial data, including encryption, firewalls, and intrusion detection systems.
- Require confirmation of each financial transaction, ensuring accuracy and minimizing errors in fund transfers
- The system should have effective error handling mechanisms to address issues promptly and prevent financial discrepancies.
- Maintain a comprehensive audit trail of all financial interactions, allowing for easy tracking and resolution of any discrepancies.
- Ensure compliance with relevant financial regulations and provide reporting capabilities for the bank to monitor its financial activities with the laundry and dry-cleaning system.

3.1.2 Non-Functional Requirements

Non-Functional Requirements simply known as quality attributes. It describes the characteristics of the system that are not directly concerned with specific functionality. Non-functional requirements may be more critical than functional requirements. If these are not achieved, the system may be useless.

Speed

- The system must have good speed.
- The system can access more users at the same time without any failures.

Availability

- The system should be available 24/7.

User-friendliness

- The system should be accessible to users with low IT literacy.

Reliability

- The system must have the ability to detect invalid user credentials.

Security

- The system should have the ability to prevent unauthorized access, misuse, forgery, and secure user data.
- Also, by providing unique user ID and password, no one can access the system by using any other's user ID and password.

Scalability

- The system should be able to handle a higher workload on-demand.

3.2 Data Requirements.

- Customer
  - C_id
  - First_Name
  - Last_Name
  - C_Type
  - C_Phone
  - Email
  - DOB

- Employee
  - Emp_id
  - First_name
  - Last_name
  - Address
  - Email
  - Age
  - DOB
  - Emp_Phone

- Manager
  - M_id
  - Sur_name
  - First_Name
  - Last_Name
  - DOB
  - Phone
  - Email
  - Address

- Dependent
  - D_ID
  - D_Name
  - Address
  - DOB

- Item
  - Item_No
  - I_Name
  - I_Type
  - I_Qty
  - I_Description

- Outlet
  - O_id
  - O_Name
  - Street
  - City
  - Postal_Code
  - O_Phone
  - O_Email

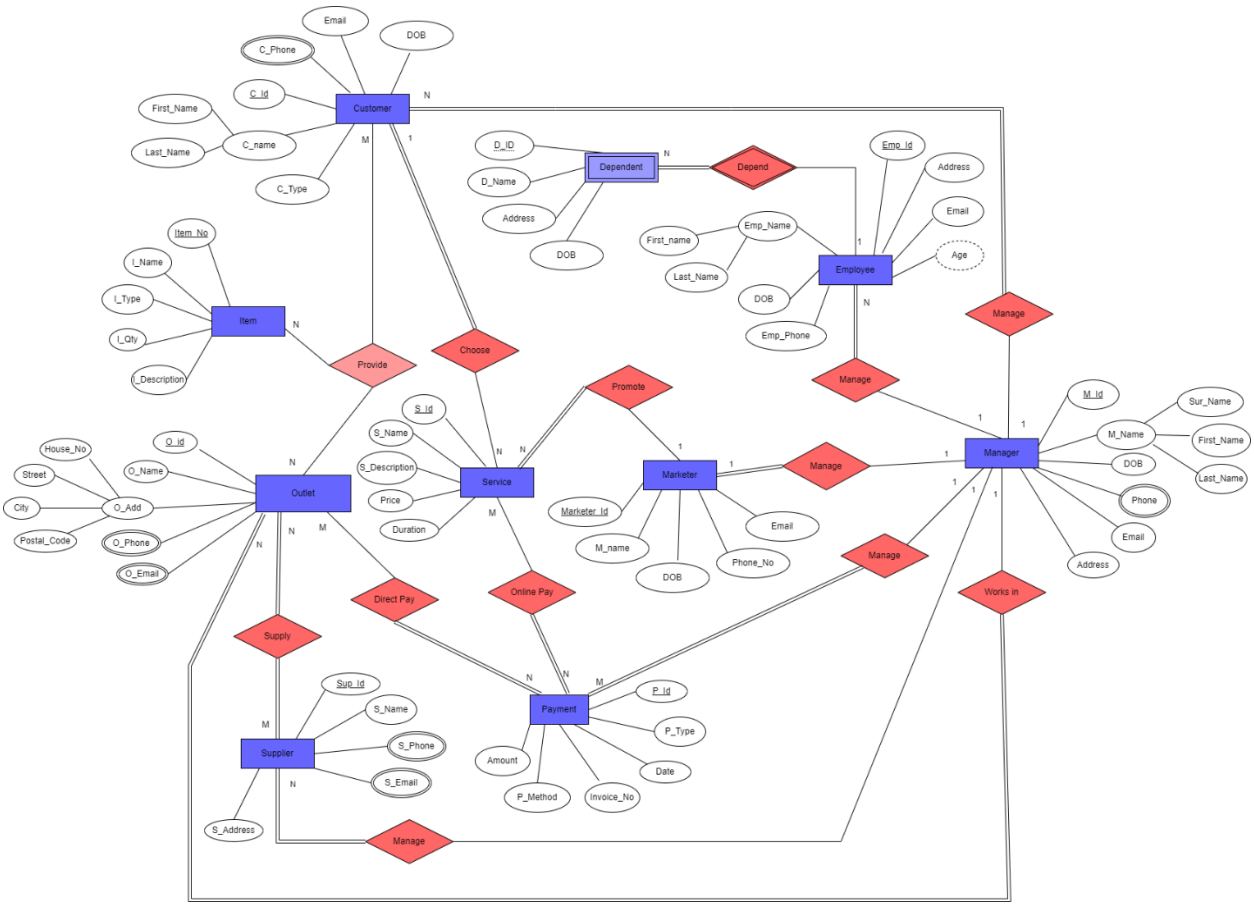
- Service
  - S_Id
  - S_name
  - S_Description
  - Price
  - Duration

- Marketer
  - Marketer_id
  - M_name
  - DOB
  - Phone_No
  - Email

- Supplier
  - S_Id
  - S_Name
  - S_Phone
  - S_Email

- Payment
  - P_id
  - P_Type
  - Date
  - Invoice_No
  - P_Method
  - Amount

3. Entity Relationship (ER Diagram).



Entity Relationship Diagram Drive Link: [ISDM Entity Relationship Diagram]

5. Relational Schema

Relational Schema Drive Link: ISDM Relational Map

6. SQL Queries

6.1.Database Creation

```
--Create table Customer
CREATE TABLE Customer(

    C_ID INT,
    Email VARCHAR(50),
    Dob DATE,
    C_Type VARCHAR(20),
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    M_Id int,

    CONSTRAINT customer_PK primary key(C_ID),
    CONSTRAINT customer_FK foreign key (M_Id) references Manager(M_Id),

);

--Create table Customer_Contact
CREATE TABLE Customer_Contact(

    C_Phone INT,
    C_ID INT,

    CONSTRAINT CC_PK primary key(C_Phone),
    CONSTRAINT CC_FK foreign key (C_ID) references Customer(C_ID),

);


 --Create table Manager
CREATE TABLE Manager(

    M_Id INT,
    Sur_Name VARCHAR(50),
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Dob DATE,
    Email VARCHAR(50),
    Address VARCHAR(100),

    CONSTRAINT Manager_pk primary key(M_Id),

);

 --Create table Manager_Contact
CREATE TABLE Manager_Contact(

    M_Id INT,
    phone INT,

    CONSTRAINT Manager_Contact_PK primary key(phone),
    CONSTRAINT Manager_Contact_FK foreign key(M_Id) references Manager(M_Id)
);


--Create table Service
CREATE TABLE Service(

    S_Id INT,
    S_Name VARCHAR(50),
    S_Description VARCHAR(100),
    Price INT,
    Duration TIME,
    Marketer_Id INT,
    C_Id INT,

    CONSTRAINT Service_pk primary key(S_Id),
    CONSTRAINT Service_Fk_Marketer foreign key(Marketer_Id) references Marketer(Marketer_id),
    CONSTRAINT Service_Fk_Customer FOREIGN KEY (C_Id) REFERENCES Customer(C_ID),
);
```

```sql
--Create table Item
CREATE TABLE Item(

    Item_No int,
    Item_Name VARCHAR(50),
    I_Type VARCHAR(50),
    I_Qty INT,
    I_Description VARCHAR(200),

    CONSTRAINT Item_Pk primary key(Item_No),

);
--Create table Outlet
CREATE TABLE Outlet(

    O_Id INT,
    O_Name VARCHAR(50),
    House_No INT,
    Street VARCHAR(50),
    City VARCHAR(50),
    Postal_Code INT,
    M_Id INT,

    CONSTRAINT Outlet_PK primary key(O_Id),
    CONSTRAINT Outlet_fk foreign key(M_Id) references Manager(M_Id),
);


 --Create table Outlet_Contact
CREATE TABLE Outlet_Contact(

     O_Id INT,
     O_Phone INT,

     CONSTRAINT Outlet_Contact_PK primary key(O_Phone),
     CONSTRAINT Outlet_Contact_FK foreign key(O_Id) references Outlet(O_Id),
 );

 --Create table Outlet_Email
CREATE TABLE Outlet_Email(

     O_Id INT,
     O_Email VARCHAR(50),

     CONSTRAINT Outlet_Email_PK primary key(O_Email),
     CONSTRAINT Outlet_Email_FK foreign key(O_Id) references Outlet(O_Id),
 );

 --Create table Provide
CREATE TABLE Provide(

    C_ID INT,
    Item_No INT,
    O_Id INT,

    CONSTRAINT Provide_FK_Outlet foreign key(O_Id) references Outlet(O_Id),
    CONSTRAINT Provide_FK_Customer  foreign key(C_ID) references Customer(C_ID),
    CONSTRAINT Provide_FK_Item foreign key(Item_No) references Item(Item_No),
 );

 --Create table Employee
CREATE TABLE Employee(

    Emp_Id INT,
    Address VARCHAR(100),
    Email VARCHAR(100),
    Age INT,
    Dob DATE,
    Emp_Phone INT,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    M_Id INT,

    CONSTRAINT Employee_PK primary key(Emp_Id),
    CONSTRAINT Employee_FK foreign key(M_Id) references Manager(M_Id),
 );
```

```sql
--Create table Dependent
CREATE TABLE Dependent(

    D_Id INT,
    Emp_id INT,
    D_Name VARCHAR(50),
    Address VARCHAR(100),
    Dob DATE,

    CONSTRAINT dependent_pk primary key(D_Id,Emp_Id),
    CONSTRAINT Dependent_FK_Employee foreign key(Emp_id) references Employee(Emp_Id),
);

--Create table Supplier
CREATE TABLE Supplier(

    Sup_Id INT,
    S_Name VARCHAR(50),
    S_Address VARCHAR(50),
    M_Id INT,

    CONSTRAINT Supplier_pk primary key(Sup_Id),
    CONSTRAINT Supplier_FK foreign key(M_Id) references Manager(M_Id),
);



--Create table Supplier_Contact
CREATE TABLE Supplier_Contact(

    Sup_Id INT,
    S_Phone CHAR(10),

    CONSTRAINT Supplier_Contact_PK primary key(S_Phone),
    CONSTRAINT Supplier_Contact_FK foreign key(Sup_Id) references Supplier(Sup_Id),
);

--Create table Suppler_Email
CREATE TABLE Supplier_Email(

    Sup_Id INT,
    S_Email VARCHAR(50),

    CONSTRAINT Supplier_Email_PK primary key(S_Email),
    CONSTRAINT Supplier_Email_FK foreign key(Sup_Id) references Supplier(Sup_Id),

);

--Create table Supply
CREATE TABLE Supply(

    O_Id INT,
    Sup_Id INT,

    CONSTRAINT Supply_FK_Outlet foreign key(O_Id) references Outlet(O_Id),
    CONSTRAINT Supply_FK_Supplier foreign key(Sup_Id) references Supplier(Sup_Id),

);

--Create table Payment
CREATE TABLE Payment(

    P_Id INT,
    P_Type VARCHAR(50),
    Date DATE,
    Invoice_No VARCHAR(50),
    P_Method VARCHAR(50),
    Amount INT,
    M_Id INT,

    CONSTRAINT Payment_PK primary key(P_Id),
    CONSTRAINT Payment_FK_Manager foreign key (M_Id) references Manager(M_Id),
);
```

```sql
--Create table Direct_Pay
CREATE TABLE Direct_Pay(

    P_Id INT,
    O_Id INT,

    CONSTRAINT Direct_Pay_FK_Payment foreign key(P_Id) references Payment(P_Id),
    CONSTRAINT Direct_Pay_FK_Outlet foreign key(O_Id) references Outlet(O_Id),

);
--Create table Online_Pay
CREATE TABLE Online_Pay(

    P_Id INT,
    S_Id INT

    CONSTRAINT Online_Pay_FK_Payment foreign key(P_Id) references Payment(P_Id),
    CONSTRAINT Online_Pay_FK_Service foreign key(S_Id) references Service(S_Id),
);


--Create table Marketer
CREATE TABLE Marketer(

    Marketer_id INT,
    M_Name VARCHAR(50),
    Dob DATE,
    Phone_No INT,
    Email VARCHAR(50),
    M_Id INT,

    CONSTRAINT Markrter_PK primary key(Marketer_id),
    CONSTRAINT Marketer_FK foreign key(M_Id) references Manager(M_Id),

);


INSERT INTO Manager
VALUES (1000, 'Smith', 'John', 'Doe', '1980-01-15', 'john.doe@email.com', '123 Main St');
```

```sql
INSERT INTO Customer
VALUES
    (1001, 'customer1@email.com', '1990-05-12', 'Daily Customer', 'John', 'Doe', 1000),
    (2002, 'customer2@email.com', '1985-03-20', 'Monthly Customer', 'Jane', 'Smith', 1000),
    (3003, 'customer3@email.com', '1982-08-15', 'Daily Customer', 'Robert', 'Johnson', 1000),
    (4004, 'customer4@email.com', '1993-07-10', 'Weekly Customer', 'Emily', 'Williams', 1000),
    (5005, 'customer5@email.com', '1980-12-28', 'Daily Customer', 'Michael', 'Brown', 1000),
    (6006, 'customer6@email.com', '1988-02-18', 'Weekly Customer', 'Sarah', 'Jones', 1000);


-- Insert data into Customer_Contact table
INSERT INTO Customer_Contact
VALUES
    (0714567890, 1001),
    (0766543210, 2002),
    (0785555555, 2002),
    (0747777777, 2002),
    (0871111111, 3003),
    (0772222222, 3003);

-- Insert data into the Manager_Contact table
INSERT INTO Manager_Contact (M_Id,Phone)
VALUES
    (1000, 0704567890),
    (1000, 0706543210);
```

```sql
-- Insert data into the Service table
INSERT INTO Service
VALUES
    (1, 'Dry Cleaning', 'Drying and Cleaning Clothes', 500, '01:30:00', 10, 1001),
    (2, 'Washing', 'Washing Clothes', 750, '02:00:00', 10, 2002),
    (3, 'Pressing', 'Pressing Clothes', 600, '01:45:00', 10, 3003),
    (4, 'Mending Service', 'Mending Clothes', 900, '02:30:00', 10, 4004),
    (5, 'Pickup and Delivery', 'Pickup and deliever clothes', 550, '01:15:00', 10, 5005);

-- Insert data into the Item table
INSERT INTO Item (Item_No, Item_Name, I_Type, I_Qty, I_Description)
VALUES
    (100, 'Trouser', 'Cotton', 10, 'White Trouser'),
    (200, 'Blouse', 'Silk', 15, 'Blue Blouse'),
    (300, 'Under Wear', 'Cotton', 20, 'Small Size Underwear'),
    (400, 'Cap', 'Cotton', 8, 'Rap Cap'),
    (500, 'Sock', 'Cotton', 12, 'Athletic Sock'),
    (600, 'Shoe', 'Leather', 5, 'Formal Shoe');


-- Insert data into the Outlet table
INSERT INTO Outlet (O_Id, O_Name, House_No, Street, City, Postal_Code, M_Id)
VALUES
    (111, 'Outlet 1', 123, 'Main St', 'City A', 12345, 1000),
    (222, 'Outlet 2', 456, 'Elm St', 'City B', 23456,1000),
    (333, 'Outlet 3', 789, 'Oak St', 'City C', 34567, 1000),
    (444, 'Outlet 4', 101, 'Pine St', 'City D', 45678, 1000),
    (555, 'Outlet 5', 222, 'Cedar St', 'City E', 56789, 1000),
    (666, 'Outlet 6', 333, 'Birch St', 'City F', 67890, 1000);

-- Insert data into the Outlet_Contact table
INSERT INTO Outlet_Contact (O_Id, O_Phone)
VALUES
    (111, 0714567890),
    (111, 0786543210),
    (222, 0705555555),
    (222, 0707777777),
    (333, 0801111111),
    (333, 0812222222);


-- Insert data into the Outlet_Email table
INSERT INTO Outlet_Email (O_Id, O_Email)
VALUES
    (111, 'outlet1@email.com'),
    (111, 'contact1@email.com'),
    (222, 'outlet2@email.com'),
    (222, 'contact2@email.com'),
    (333, 'outlet3@email.com'),
    (333, 'contact3@email.com');


-- Insert data into the Provide table
INSERT INTO Provide (C_ID, Item_No, O_Id)
VALUES
    (1001, 100, 111),
    (2002, 200, 222),
    (3003, 300, 333),
    (4004, 400, 444),
    (5005, 500, 555),
    (6006, 600, 666);
```

```sql
-- Insert data into the Employee table
INSERT INTO Employee (Emp_Id, Address, Email, Age, Dob, Emp_Phone, First_Name, Last_Name, M_Id)
VALUES
    (1010, '123 Main St', 'employee1@email.com', 30, '1990-01-15', 0704567890, 'John', 'Doe', 1000),
    (2020, '456 Elm St', 'employee2@email.com', 28, '1992-03-20', 0706543210, 'Jane', 'Smith', 1000),
    (3030, '789 Oak St', 'employee3@email.com', 35, '1985-05-10', 0755555555, 'Robert', 'Johnson', 1000),
    (4040, '101 Pine St', 'employee4@email.com', 25, '1995-08-28', 0917777777, 'Emily', 'Williams', 1000),
    (5050, '222 Cedar St', 'employee5@email.com', 32, '1988-02-12', 0861111111, 'Michael', 'Brown', 1000),
    (6060, '333 Birch St', 'employee6@email.com', 27, '1993-06-17', 0862222222, 'Sarah', 'Jones', 1000);

-- Insert data into the Dependent table
INSERT INTO Dependent (D_Id, Emp_id, D_Name, Address, Dob)
VALUES
    (101, 1010, 'Dileepa', '123 Main St', '2010-03-05'),
    (202, 2020, 'Sarithmal', '456 Elm St', '2012-07-10'),
    (303, 3030, 'pasindu', '789 Oak St', '2008-01-15');


-- Insert data into the Supplier table
INSERT INTO Supplier (Sup_Id, S_Name, S_Address, M_Id)
VALUES
    (110, 'Gunapala', '123 Supplier St', 1000),
    (220, 'KFC', '456 Supplier Rd', 1000),
    (330, 'Ayubowan', '789 Supplier Ave', 1000),
    (440, 'Nona Haamu', '101 Supplier Blvd', 1000),
    (660, 'David', '333 Supplier Dr', 1000);



-- Insert data into the Supplier_Contact table
INSERT INTO Supplier_Contact (Sup_Id, S_Phone)
VALUES
    (110, 0701111111),
    (110, 0782222222),
    (220, 0763333333),
    (220, 0984444444),
    (330, 0775555555),
    (330, 0756666666);

-- Insert data into the Supplier_Email table
INSERT INTO Supplier_Email (Sup_Id, S_Email)
VALUES
    (110, 'supplier1@email.com'),
    (110, 'contact1@email.com'),
    (220, 'supplier2@email.com'),
    (220, 'contact2@email.com'),
    (330, 'supplier3@email.com'),
    (330, 'contact3@email.com');


-- Insert data into the Supply table
INSERT INTO Supply(O_Id, Sup_Id)
VALUES
    (111, 110),
    (222, 220),
    (333, 330),
    (444, 440),
    (555, 440),
    (666, 330);

-- Insert data into the Payment table
INSERT INTO Payment (P_Id, P_Type, Date, Invoice_No, P_Method, Amount, M_Id)
VALUES
    (10001, 'E Money', '2023-01-15', 'INV001', 'Online Payment', 1500, 1000),
    (20001, 'E Money', '2023-02-20', 'INV002', 'Visa', 1500, 1000),
    (30001, 'E Money', '2023-03-10', 'INV003', 'Master', 1200, 1000),
    (40001, 'E Money', '2023-05-12', 'INV004', 'American Express', 1100, 1000),
    (50001, 'E Money', '2023-06-17', 'INV005', 'Online Pyment', 1250, 1000),
    (60001, 'Cash', '2023-06-17', 'INV006', 'ATM', 1700, 1000);
```

```sql
-- Insert data into the Direct_Pay table
INSERT INTO Direct_Pay (P_Id, O_Id)
VALUES
    (10001, 111),
    (20001, 222),
    (30001, 333),
    (40001, 444),
    (50001, 555),
    (60001, 666);

-- Insert data into the Online_Pay table
INSERT INTO Online_Pay (P_Id, S_Id)
VALUES
    (10001, 1),
    (20001, 2),
    (30001, 3),
    (40001, 4),
    (50001, 5),
    (60001, 5);

-- Insert data into the Marketer table
INSERT INTO Marketer (Marketer_id, M_Name, Dob, Phone_No, Email, M_Id)
VALUES
    (10, 'Ravindu', '1990-01-15', 0704567890, 'Ravindu@email.com', 1000);
```

Customer:

	C_ID	Email	Dob	C_Type	First_Name	Last_Name	M_Id
1	1001	customer1@email.com	1990-05-12	Daily Customer	John	Doe	1000
2	2002	customer2@email.com	1985-03-20	Monthly Customer	Jane	Smith	1000
3	3003	customer3@email.com	1982-08-15	Daily Customer	Robert	Johnson	1000
4	4004	customer4@email.com	1993-07-10	Weekly Customer	Emily	Williams	1000
5	5005	customer5@email.com	1980-12-28	Daily Customer	Michael	Brown	1000
6	6006	customer6@email.com	1988-02-18	Weekly Customer	Sarah	Jones	1000

Customer_Contact:

	C_Phone	C_ID
1	714567890	1001
2	747777777	2002
3	766543210	2002
4	772222222	3003
5	785555555	2002
6	871111111	3003

Manager:

	M_Id	Sur_Name	First_Name	Last_Name	Dob	Email	Address
1	1000	Smith	John	Doe	1980-01-15	john.doe@email.com	123 Main St

Manager_Contact:

	M_Id	phone
1	1000	704567890
2	1000	706543210

Service:

	S_Id	S_Name	S_Description	Price	Duration	Marketer_Id	C_Id
1	1	Dry Cleaning	Drying and Cleaning Clothes	500	01:30:00.0000000	10	1001
2	2	Washing	Washing Clothes	750	02:00:00.0000000	10	2002
3	3	Pressing	Pressing Clothes	600	01:45:00.0000000	10	3003
4	4	Mending S...	Mending Clothes	900	02:30:00.0000000	10	4004
5	5	Pickup an...	Pickup and deliever clothes	550	01:15:00.0000000	10	5005

Item:

	Item_No	Item_Name	I_Type	I_Qty	I_Description
1	100	Trouser	Cotton	10	White Trouser
2	200	Blouse	Silk	15	Blue Blouse
3	300	Under Wear	Cotton	20	Small Size Underwear
4	400	Cap	Cotton	8	Rap Cap
5	500	Sock	Cotton	12	Athletic Sock
6	600	Shoe	Leather	5	Formal Shoe

Outlet:

Results Messages

	O_Id	O_Name	House_No	Street	City	Postal_Code	M_Id
1	111	Outlet 1	123	Main St	City A	12345	1000
2	222	Outlet 2	456	Elm St	City B	23456	1000
3	333	Outlet 3	789	Oak St	City C	34567	1000
4	444	Outlet 4	101	Pine St	City D	45678	1000
5	555	Outlet 5	222	Cedar St	City E	56789	1000
6	666	Outlet 6	333	Birch St	City F	67890	1000

Outlet_Contact:

	O_Id	O_Phone
1	222	705555555
2	222	707777777
3	111	714567890
4	111	786543210
5	333	801111111
6	333	812222222

Outlet_Email:

	O_Id	O_Email
1	111	contact1@email.com
2	222	contact2@email.com
3	333	contact3@email.com
4	111	outlet1@email.com
5	222	outlet2@email.com
6	333	outlet3@email.com

Provide:

	C_ID	Item_No	O_Id
1	1001	100	111
2	2002	200	222
3	3003	300	333
4	4004	400	444
5	5005	500	555
6	6006	600	666

Employee:

	Emp_Id	Address	Email	Age	Dob	Emp_Phone	First_Name	Last_Name	M_Id
1	1010	123 Main St	employee1@email.com	30	1990-01-15	704567890	John	Doe	1000
2	2020	456 Elm St	employee2@email.com	28	1992-03-20	706543210	Jane	Smith	1000
3	3030	789 Oak St	employee3@email.com	35	1985-05-10	755555555	Robert	Johnson	1000
4	4040	101 Pine St	employee4@email.com	25	1995-08-28	917777777	Emily	Williams	1000
5	5050	222 Cedar St	employee5@email.com	32	1988-02-12	861111111	Michael	Brown	1000
6	6060	333 Birch St	employee6@email.com	27	1993-06-17	862222222	Sarah	Jones	1000

Dependent:

	D_Id	Emp_id	D_Name	Address	Dob
1	101	1010	Dileepa	123 Main St	2010-03-05
2	202	2020	Sarithmal	456 Elm St	2012-07-10
3	303	3030	pasindu	789 Oak St	2008-01-15

Supplier:

	Sup_Id	S_Name	S_Address	M_Id
1	110	Gunapala	123 Supplier St	1000
2	220	KFC	456 Supplier Rd	1000
3	330	Ayubowan	789 Supplier Ave	1000
4	440	Nona Haamu	101 Supplier Blvd	1000
5	660	David	333 Supplier Dr	1000

Supplier_Contact:

	Sup_Id	S_Phone
1	110	1111111111
2	110	2222222222
3	220	3333333333
4	220	4444444444
5	330	5555555555
6	330	6666666666

Supplier_Email:

	Sup_Id	S_Email
1	110	contact1@email.com
2	220	contact2@email.com
3	330	contact3@email.com
4	110	supplier1@email.com
5	220	supplier2@email.com
6	330	supplier3@email.com

Supply:

	O_Id	Sup_Id
1	111	110
2	222	220
3	333	330
4	444	440
5	555	440
6	666	330

Marketer:

	Marketer_id	M_Name	Dob	Phone_No	Email	M_Id
1	10	Ravindu	1990-01-15	704567890	Ravindu@email.com	1000

Payment:

	P_Id	P_Type	Date	Invoice_No	P_Method	Amount	M_Id
1	10001	E Money	2023-01-15	INV001	Online Payment	1500	1000
2	20001	E Money	2023-02-20	INV002	Visa	1500	1000
3	30001	E Money	2023-03-10	INV003	Master	1200	1000
4	40001	E Money	2023-05-12	INV004	American Express	1100	1000
5	50001	E Money	2023-06-17	INV005	Online Payment	1250	1000
6	60001	Cash	2023-06-17	INV006	ATM	1700	1000

Online_Pay:

	P_Id	S_Id
1	10001	1
2	20001	2
3	30001	3
4	40001	4
5	50001	5
6	60001	5

Direct_Pay:

	P_Id	O_Id
1	10001	111
2	20001	222
3	30001	333
4	40001	444
5	50001	555
6	60001	666

7. Performance Requirements

A major role is played by Performance Requirements to make the system successful. They are as follow,

- The system must be active 24 hours, 365 days for a customer to access the system without any inconvenience.
- A Registered User can access the system numerous times by entering his/ her login credentials.
- The login process and loading the pages must be done within a few seconds.
- Speed and Usability are the performance requirements for this system.
- Registered users can view service details.
- Registered users can edit or delete his/her account details.
- System loads within a minimum time.
- Administrators can add or remove shared experiences and feedback, manage User accounts etc.
- System allows Dry Cleaners to respond to the Users messages.
- Design user friendly user interface.
- Users must be able to access the website at any time using any device or browser.
- System provides ability for Administrator to manage the staff account.

8. Security Requirements

- Personal details of users should be encrypted before sending to the database.
- Unauthorized users should be unable to access restricted features.
- The database should have a backup of all the data in the system.
- The password of a user account must be a strength password which includes uppercase letters, lowercase letters, numbers, and special characters.
- Database server must be maintained without redundancy.
- Only the administrators can access and modify the data of the system.