

Information Systems and Data Modeling – IT1090



Assignment

Title: Online Customer Support System

Batch Number: 01

Group Number: MLB_01.02_07

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:

IT22915818

Dilshan B A T

.....
signature 

IT22890528

Wickramasuriya C I

.....
signature 

IT22345578

Galappaththi A G R S

.....
signature 

IT22362094

Wijesiri W G S S

.....
signature 

IT21xxxxxx

<name>

.....
signature

signature

Submitted on: 27/05/2023

1. Introduction

It has become a huge trend to shop through e-commerce platforms, and it helps customers save time and explore as many items as they want efficiently. Because of that, e-commerce platforms have become an impressive option for customers to do their shopping. A customer support system is a type of system that is crucial for the existence of an e-commerce platform. An online customer support system helps its main business to retain its customers by helping customers to resolve their inquiries. Through a customer support system, a customer can resolve any problem. This system provides features like live chat, online ticket raising, FAQs, emails, phone calls, and social media platforms. With those features, a customer can easily reach institutional support.

This system uses programming languages like HTML, CSS, JavaScript, PHP, and My SQL. When it comes to databases used in this system, it is necessary to have them. Because it is essential to store data that the system gathers regarding customers' inquiries, it helps the business identify what type of problems customers face and take necessary actions to help customers by resolving their inquiries. Those can be crucial for improving and enhancing the business's efficiency.

There are two main aspects of requirement analysis. Those are functional requirements and non-functional requirements. A functional requirement is what happens in the system according to users' requirements. In contrast, non-functional requirements are requirements that do not affect the direct uses of the system but are necessary to improve the system's efficiency. Non-functional requirements include efficiency, reliability, user-friendliness, data backup system, understanding, and responsibility. Through the requirement analysis phase, all the requirements that should be contained in the system should be gathered. Interviews, questionnaires, observation, prototyping, existing documents, and report studies can be used to obtain those requirements.

1. Hypothetical Scenario

The online customer support system is an online platform that supports customers of an e-commerce platform to resolve their inquiries regarding their experiences in the e-commerce platform. This system provides customers with a wide range of features to satisfy them by providing a quality service. Guest Users, Registered customers, and sellers can efficiently resolve problems through this system.

The system is accessible for registered and unregistered users (guest users). Registered users can access all the features that a guest user can access. A guest user can access the FAQ section, register option and view the feedback option. Registered users can use the contact us feature, reset password option, login option, view their profiles, and check their returns, orders, payment options, and loyalty points. Through this system, a seller can access refund details, return details, Q & A section, view their profile, reset password and check customer service agents' reports.

Each feature has its sub-systems. When it comes to the contact us option, it provides a variety of options to reach a customer service agent. It involves sending emails, raising tickets, live chat, phone calls, and social media platforms. When a customer or a seller registers with the system, it generates profiles for individuals. From registration onwards, a customer or a seller can view their profile and reset their password. Furthermore, registered customer can check their returns history, order history, and check their loyalty points. In contrast, a seller can obtain their customer's refund and return details and engage with them through the Q & A section.

The system includes administrative roles such as customer service agent, system admin, and manager to handle the platform's many components efficiently. Customer service agents communicate with customers and sellers and manage their feedback, while the system admin manages staff accounts. Moreover, the system admin can access anything except the work the manager can do. The manager holds higher-level access to the system and engages in generating reports.

2. Requirement Analysis

2.1 Main Requirement

3.1.1 Functional Requirements

1. Guest User

User requirements-

- Guest users can register in the system by providing the required information for registration.
- Guest users can register as a seller or customer.
- Guest users should be assured that their personal information will be handled securely and in accordance with applicable data protection regulations.
- Guest users should have access to a knowledge base or FAQ section to find answers to common questions and issues.
- Guest users can view the available Feedbacks.
- Feedback from guest users can be collected anonymously or by providing an optional email address.

System requirements-

- If guest users choose to register, the system should provide a streamlined and user-friendly registration process.
- The system should provide an option for guest users to register for an account, but it should not be mandatory.
- The system should provide clear instructions and validation checks to ensure correct and valid registration information.
- The system should ensure the privacy and protection of guest user data and information.
- The system must approve the registration details and create a registered customer account.
- After registration, Guest Users should receive a confirmation email or notification to verify their email address and activate their account.

2.Registerd customer

User requirements-

- Registered customer can login to the system by providing the required user login credentials.
- Registered customer should be able to easily manage their account details, including personal information, contact preferences and login credentials.
- They should be able to update their profiles, change passwords and manage communication preferences.
- Registered customers must have the ability to submit support tickets through the system.
- Registered customers should be able to easily initiate and continue conversations with support agents using their preferred communication channel.
- Registered customers should have the opportunity to provide feedback on their support experience and evaluate the quality of service received.
- Registered customers can check orders and returns.

System requirements-

- The system should provide mechanisms to help registered customers share their feedback, suggestions, or complaints to help improve the system.
- The system must have appropriate authentication and authorization mechanisms to ensure the security of registered user accounts.
- The system must have a ticket management module to handle support tickets submitted by registered users.
- The system should provide multiple communication channels for registered users to interact with support agents.
- Common communication channels may include email, live chat, phone support, or a dedicated messaging system within the support system.
- The system should store/save the updated customer account details.

3.Seller

User requirements-

- Sellers should be able to register and set up their profile.
- Sellers should be able to update their account information and manage their account settings.
- The seller can access Q & A.
- Sellers should be able to manage and update information related to their products or services in the support system.
- Sellers should have access to reports on ticket volume, response time, resolution time, customer satisfaction ratings and other relevant metrics.
- Sellers should be able to create and update articles, FAQs, or other support resources related to their products or services.

System requirements-

- The system should provide a user-friendly interface for sellers to create and manage their accounts.
- The system should allow sellers to manage their account settings, including profile updates, password changes and notification preferences.
- The system should support multiple communication channels for salespeople to interact with customers.
- The system should provide reporting and analysis features to help salespeople track and analyze customer support data.
- Sellers need seamless access to customer information, order history and other relevant data to provide comprehensive support.

4. Customer Service Agent

User requirements-

- Customer service agents can manage feedback.
- Customer service agents should manage social media platforms.
- Agents should have quick access to relevant customer information, including order history, previous interactions, and ongoing issues. This helps agents provide personalized and efficient support.
- Agents should be able to easily manage and update ticket statuses, ensuring timely and efficient resolution.

System requirements-

- If the customer base is diverse, the system should support different languages for customers from different regions.
- The system should enable agents to access customer information easily.
- The system should include a ticketing system that allows agents to create, assign, track, and prioritize customer inquiries.
- The system must be able to handle increasing volumes of customer inquiries and support an increasing number of agents.

5. System Admin

User requirements-

- System admin can manage staff accounts.
- The system admin must have the ability to manage user accounts, including creating new accounts, changing user roles and permissions, and disabling or deleting accounts when necessary.
- They should also be able to handle password resets and account troubleshooting.
- The system administrator is responsible for managing the security measures in the system.
- The system administrator is responsible for managing system upgrades and maintenance.

System requirements-

- The system should provide user management capabilities for the administrator.
- The system should provide reporting and analytics features to enable the system administrator to track key performance metrics, generate reports on system usage, user activity, and other relevant data.

6. Manager

User requirements-

- Managers can generate reports.
- A manager must have strong leadership skills to effectively lead and motivate the support team.
- A manager must be adept at analyzing complex problems, identifying root causes, and implementing effective solutions to improve customer satisfaction.
- Excellent written and verbal communication skills are essential for a customer support manager.
- Managers should be able to provide feedback, conduct audits, and identify areas for improvement in agent performance.
- Prior experience in customer service is essential. A manager must have a deep understanding of customer needs and expectations, be familiar with common support challenges, and be adept at handling difficult situations.

System requirements-

- The system should support effective team management.
- The system should provide a user-friendly and intuitive interface for managers to access and navigate various features and functions.
- The system should facilitate performance monitoring of agents, enabling managers to monitor their activities, response times and customer interactions.
- The system should support quality assurance processes that allow managers to review and evaluate customer interactions, such as chat transcripts or recorded calls.
- The system should allow managers to set performance goals for individual agents or the entire team.
- The system should allow managers to configure and manage support workflows and processes.

3.1.2 Non-Functional Requirements

Non-functional requirements are requirements that describe criteria rather than particular behaviors that may be used to evaluate how well a system performs. Non-functional requirements of an online customer support system are typically based on the system's performance, security, usability, dependability, and scalability.

Non-functional requirements of an online customer support system include the following:

1. **Performance:**
For guaranteed on-time customer support interactions, the system should offer quick responses. For instance, it should strive to load sites quickly and reduce latency during chat sessions.
2. **Security:**
To safeguard client data and guarantee its confidentiality, integrity, and availability, the system should put in place strong security measures. This involves conforming to pertinent data protection laws, secure user authentication, and encrypted communication methods.
3. **Usability:**
Both clients and support staff should find the system simple to use. In order to support successful customer interactions, it should have a user-friendly design, simple navigation, and offer important guidance.
4. **Reliability:**
Customers should always be able to access the system, which should be extremely reliable. This include keeping downtime to a minimum, making sure that data backups are made, and putting a recovery plan in the event in place to quickly resume service in the event of a disruption.
5. **Scalability:**
The system has to be able to handle increased client demand without strongly reducing its performance. In order to support more customers and handle increased user traffic, it should be developed to scale both horizontally and vertically.
6. **Accessibility:**
To ensure that all users, regardless of ability, can access the support services, the system must be accessible to people with disabilities and must follow accessibility rules.
7. **Performance monitoring:**
In order to watch system performance, find bottlenecks, and collect data for analysis and optimization, the system should have built-in monitoring capabilities. Functionalities like reporting, analytics, and logging may be added.
8. **Integration:**
To simplify support processes and offer a smooth experience for customers, the system should be able connect to other pertinent systems and tools, such as customer relationship management systems or knowledge bases.

2.2 Data Requirement

Guest User

Guest User ID

Name

Seller Registration

Seller User Name

Password

E mail

Feedback

Customer ID

Customer Name

Feedback

Rate

FAQ

Q No

Question

Answer

Customer Registration

Customer User Name

Customer Password

Email

Customer Service Argent

Customer Service Argent ID

Name

Password

E mail

Contact Number

Registered Customer

Customer ID

Name

Password

E mail

Contact Number

Seller

Seller ID

Name

Password

Email

Contact Number

Social Media

Account Name

Password

Type

Customer Password

Customer ID

Current Password

New Password

Customer Profile

Customer ID

Name

Email

Password

Address

Contact Number

DOB

Gender

Age

Payment Option

Order ID

Price

Payment Method

Order

Order ID

Order

Price

Address

Contact Number

E mail

Return

Return ID

Order ID

Order

Price

Address

Contact Number

E mail

Loyalty Point

Refund Details

Return ID

Price

Seller Profile

Seller ID

Name

Email

Password

Address

Contact Number

Seller Password

Seller ID

Current Password

New Password

Report

Report Number

Title

Manage

Manage ID

Name

Email

Contact Number

System Admin

Admin ID

Name

Email

Contact Number

Customer Account

Customer ID

Customer Name

Situation

Seller Account

Seller ID

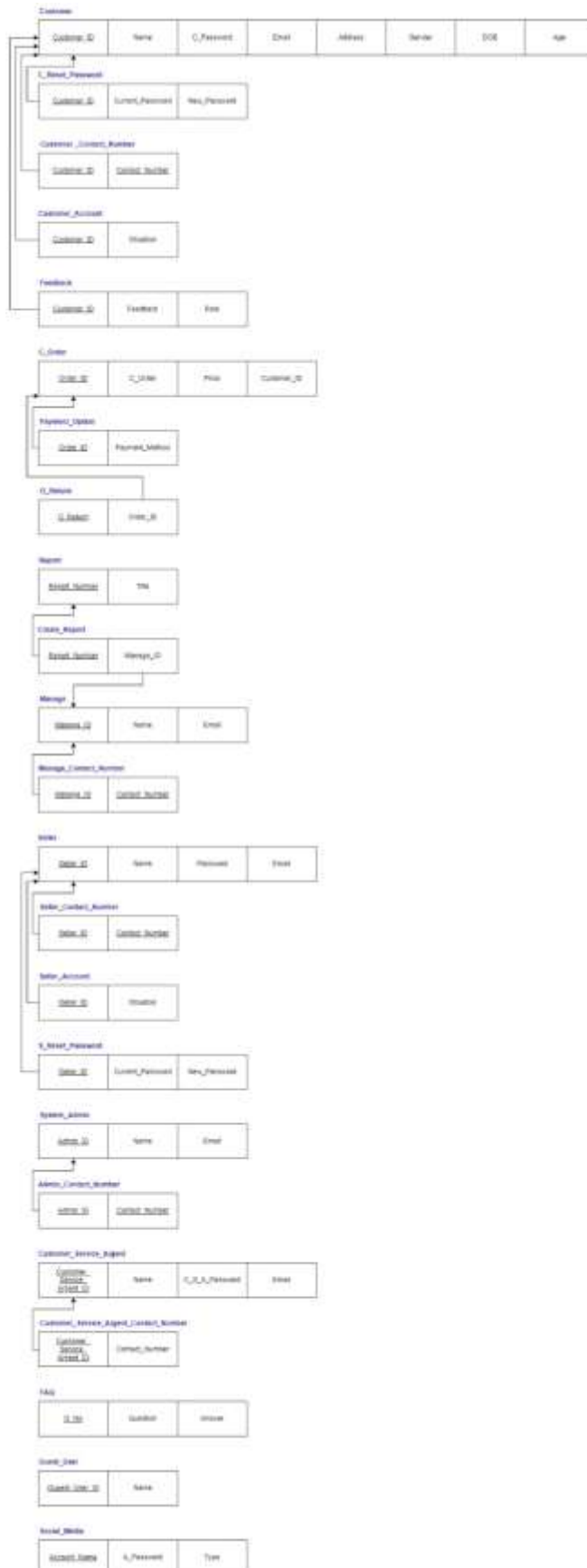
Name

Situation

The ER diagram illustrates the database structure for the system. It features several main entities and their relationships:

- Guest User**: Attributes include Guest User ID, Name, Password, E-mail, and Address. Relationships: 'is a' with Admin, 'has' with Sales Representative, and 'knows' with Customer.
- Admin**: Attributes include Admin ID, Name, Password, E-mail, and Address. Relationship: 'is a' with Guest User.
- Sales Representative**: Attributes include Sales Representative ID, Name, Password, E-mail, and Address. Relationship: 'has' with Guest User.
- Customer**: Attributes include Customer ID, Name, Password, E-mail, and Address. Relationships: 'knows' with Guest User and 'has' with various roles.
- Roles**: Includes Customer Service Agent, Account Manager, and Manager. Each role has specific attributes like Name, Password, E-mail, and Address.
- Relationships**: 'is a' (Guest User to Admin), 'has' (Guest User to Sales Representative), 'knows' (Guest User to Customer), and 'has' (Customer to various roles).

5. Relationship Scheme



6. SQL Queries

6.1 Data Base Create

```
CREATE TABLE Guest_User (  
    Guest_User_ID int NOT NULL,  
    Guest_User_Name varchar(30),  
    PRIMARY KEY (Guest_User_ID)  
)
```

```
CREATE TABLE Seller (  
    Seller_ID int NOT NULL,  
    Name varchar(30),  
    Seller_Password varchar(20),  
    Email varchar(30),  
    PRIMARY KEY (Seller_ID)  
)
```

```
CREATE TABLE Feedback (  
    Customer_ID int NOT NULL,  
    Feedback text,  
    Rate int,  
    PRIMARY KEY (Customer_ID)  
)
```

```
CREATE TABLE FAQ (  

```

```
Q_No int NOT NULL,  
Question text,  
Answer text,  
PRIMARY KEY(Q_No)  
)
```

```
CREATE TABLE Seller_Contact_Number (  
    Seller_ID int NOT NULL,  
    Contact_Number int(10),  
    PRIMARY KEY (Seller_ID,Contact_Number),  
    FOREIGN KEY (Seller_ID) REFERENCES Seller(Seller_ID)  
)
```

```
CREATE TABLE Customer_Service_Argent (  
    Customer_Service_Argent_ID int NOT NULL,  
    Name varchar(30),  
    C_S_A_Password varchar(20),  
    Email varchar(30),  
    PRIMARY KEY (Customer_Service_Argent_ID)  
)
```

```
CREATE TABLE Customer_Service_Argent_Contact_Number (  
    Customer_Service_Argent_ID int NOT NULL,  
    Contact_Number int (10),  
    PRIMARY KEY (Customer_Service_Argent_ID,Contact_Number),
```

```
FOREIGN KEY (Customer_Service_Argent_ID) REFERENCES  
Customer_Service_Argent(Customer_Service_Argent_ID)  
)
```

```
CREATE TABLE Customer (  
    Customer_ID int NOT NULL,  
    Name varchar(30),  
    C_Password varchar(20),  
    Email varchar(30),  
    Address text,  
    Gender varchar(10),  
    DOB date,  
    Age int,  
    PRIMARY KEY (Customer_ID)  
)
```

```
CREATE TABLE Customer_Contact_Number (  
    Customer_ID int,  
    Contact_Number int(10),  
    PRIMARY KEY (Customer_ID,Contact_Number),  
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)  
)
```

```
CREATE TABLE Social_Media (  
    Account_Name varchar(20) NOT NULL,
```

```
A_Password varchar(20),  
Type varchar(20),  
PRIMARY KEY (Account_Name)  
)
```

```
CREATE TABLE C_Reset_Password (  
    Customer_ID int NOT NULL,  
    Current_Password varchar(20),  
    New_Password varchar(20),  
    PRIMARY KEY (Customer_ID),  
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)  
)
```

```
CREATE TABLE C_Order (  
    Order_ID int NOT NULL,  
    C_Order text,  
    Price int,  
    Customer_ID int,  
    PRIMARY KEY (Order_ID),  
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)  
)
```

```
CREATE TABLE S_Reset_Password (  
    Seller_ID int NOT NULL,  
    Current_Password varchar(20),
```

```
Reset_Password varchar(20),  
PRIMARY KEY (Seller_ID),  
FOREIGN KEY (Seller_ID) REFERENCES Seller(Seller_ID)  
)
```

```
CREATE TABLE Payment_Option (  
    Order_ID int NOT NULL,  
    Payment_Method varchar(10),  
    PRIMARY KEY (Order_ID),  
    FOREIGN KEY (Order_ID) REFERENCES C_Order(Order_ID)  
)
```

```
CREATE TABLE O_Return (  
    O_Return int,  
    Order_ID int,  
    PRIMARY KEY (O_Return),  
    FOREIGN KEY (Order_ID) REFERENCES C_Order(Order_ID)  
)
```

```
CREATE TABLE Manage (  
    Manage_ID int NOT NULL,  
    Name varchar (20),  
    Email varchar(30),  
    PRIMARY KEY (Manage_ID)  
)
```

```
CREATE TABLE Manage_Contact_Number (  
    Manage_ID int,  
    Contact_Number int(10),  
    PRIMARY KEY (Manage_ID,Contact_Number),  
    FOREIGN KEY (Manage_ID) REFERENCES Manage(Manage_ID)  
)
```

```
CREATE TABLE Report (  
    Report_Number int NOT NULL,  
    Title varchar(20),  
    PRIMARY KEY (Report_Number)  
)
```

```
CREATE TABLE Creat_Report (  
    Report_Number int,  
    Manage_ID int,  
    PRIMARY KEY (Report_Number,Manage_ID),  
    FOREIGN KEY (Report_Number) REFERENCES Report(Report_Number),  
    FOREIGN key (Manage_ID) REFERENCES Manage(Manage_ID)  
)
```

```
CREATE TABLE System_Admin (  
    Admin_ID int NOT NULL,  
    Name varchar(20),
```

```
Email varchar(30),  
PRIMARY KEY (Admin_ID)  
)
```

```
CREATE TABLE Admin_Contact_Number (  
Admin_ID int,  
Contact_Number int(10),  
PRIMARY KEY (Admin_ID, Contact_Number),  
FOREIGN KEY (Admin_ID) REFERENCES System_Admin (Admin_ID)  
)
```

```
CREATE TABLE Customer_Account (  
Customer_ID int,  
Situation varchar(20),  
PRIMARY KEY (Customer_ID),  
FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)  
)
```

```
CREATE TABLE Sellr_Account (  
Seller_ID int,  
Situation varchar(20),  
PRIMARY KEY (Seller_ID),  
FOREIGN KEY (Seller_ID) REFERENCES Seller(Seller_ID)  
)
```

6.2 Data Store in Data Base

```
INSERT INTO Customer (Customer_ID, Name, C_Password, Email, Address, Gender, DOB, Age)
```

```
VALUES (1001, 'Namal', 'Namal@2001', 'Namal@gmail.com', '5D,Temples Road,Kaduwela,Galle.', 'Male', 1999-02-05, 24);
```

```
INSERT INTO Customer (Customer_ID, Name, C_Password, Email, Address, Gender, DOB, Age)
```

```
VALUES (1002, 'Kamal', 'Kamal@1998', 'kamal@gmail.com', '5D,Dewasurendra Mawatha,Kaluwela,Galle.', 'Male', 1998-08-11, 25);
```

```
INSERT INTO Customer (Customer_ID, Name, C_Password, Email, Address, Gender, DOB, Age)
```

```
VALUES (1002, 'Shamali', 'Shamali@2000', 'shamali@gmail.com', '5D,Isurupura,Kaluwela,Galle.', 'Female', 2000-05-24, 23);
```

```
INSERT INTO Customer (Customer_ID, Name, C_Password, Email, Address, Gender, DOB, Age)
```

```
VALUES (1004, 'Kamali', 'Kamali@1998', 'kamali@gmail.com', '5D,Mathsiri Mawatha,Kaluwela,Galle.', 'Female', 1998-10-26, 25);
```

```
INSERT INTO Customer (Customer_ID, Name, C_Password, Email, Address, Gender, DOB, Age)
```

```
VALUES (1005, 'Pasindu', 'Pasindu@1996', 'pasindu@gmail.com', '5D,Isurupura,Kaluwela,Galle.', 'Male', 1996-07-02, 27);
```

```
INSERT INTO Customer_Account (Customer_ID, Situation)
```

```
VALUES (1001, 'Activated');
```



```
INSERT INTO Customer_Account (Customer_ID, Situation)
VALUES (1002, 'Deactivated');
```

```
INSERT INTO Customer_Account (Customer_ID, Situation)
VALUES (1003, 'Deactivated');
```

```
INSERT INTO Customer_Account (Customer_ID, Situation)
VALUES (1004, 'Activated');
```

```
INSERT INTO Customer_Account (Customer_ID, Situation)
VALUES (1005, 'Activated');
```

```
INSERT INTO Customer_Contact_number (Customer_ID, Contact_Number)
VALUES (1001, 0714586523);
```

```
INSERT INTO Customer_Contact_number (Customer_ID, Contact_Number)
VALUES (1002, 0716357297);
```

```
INSERT INTO Customer_Contact_number (Customer_ID, Contact_Number)
VALUES (1003, 0722687495);
```

```
INSERT INTO Customer_Contact_number (Customer_ID, Contact_Number)
VALUES (1003, 0726963892);
```

```
INSERT INTO Customer_Contact_number (Customer_ID, Contact_Number)
VALUES (1004, 0722662453);
```

```
INSERT INTO Customer_Contact_number (Customer_ID, Contact_Number)
VALUES (1005, 0786349528);
```

```
INSERT INTO Customer_Contact_number (Customer_ID, Contact_Number)
VALUES (1005, 07863667528);
```

```
INSERT INTO Customer_Contact_number (Customer_ID, Contact_Number)
VALUES (1005, 0786967138);
```

```
INSERT INTO customer_service_argent (Customer_Service_Argent_ID, Name,
C_S_A_Password, Email)
VALUES (101, 'Nalin', 'Nalin#2023', 'nalin@gmail.com');
```

```
INSERT INTO customer_service_argent (Customer_Service_Argent_ID, Name,
C_S_A_Password, Email)
VALUES (102, 'Rashini', 'Rashini#2020', 'rashini@gmail.com');
```

```
INSERT INTO customer_service_argent (Customer_Service_Argent_ID, Name,
C_S_A_Password, Email)
VALUES (103, 'Shehan', 'Shehan#2021', 'shehan@gmail.com');
```

```
INSERT INTO customer_service_argent (Customer_Service_Argent_ID, Name,
C_S_A_Password, Email)
VALUES (104, 'Naomi', 'Naomi#2021', 'naomi@gmail.com');
```

```
INSERT INTO customer_service_argent (Customer_Service_Argent_ID, Name,  
C_S_A_Password, Email)
```

```
VALUES (105, 'Pavan', 'Pavan#2018', 'pavan@gmail.com');
```

```
INSERT INTO customer_service_argent_contact_number (Customer_Service_Argent_ID,  
Contact_Number)
```

```
VALUES (101, 0769543256);
```

```
INSERT INTO customer_service_argent_contact_number (Customer_Service_Argent_ID,  
Contact_Number)
```

```
VALUES (101, 0719365367);
```

```
INSERT INTO customer_service_argent_contact_number (Customer_Service_Argent_ID,  
Contact_Number)
```

```
VALUES (102, 0768024645);
```

```
INSERT INTO customer_service_argent_contact_number (Customer_Service_Argent_ID,  
Contact_Number)
```

```
VALUES (103, 0716387459);
```

```
INSERT INTO customer_service_argent_contact_number (Customer_Service_Argent_ID,  
Contact_Number)
```

```
VALUES (103, 0716987450);
```

```
INSERT INTO customer_service_argent_contact_number (Customer_Service_Argent_ID,  
Contact_Number)
```

```
VALUES (104, 0763822055);
```

```
INSERT INTO customer_service_argent_contact_number (Customer_Service_Argent_ID,  
Contact_Number)
```

```
VALUES (105, 0719823668);
```

```
INSERT INTO C_Order (Order_ID, C_Order, Price, Customer_ID)
```

```
VALUES (10001, 'Book-12, Pen-10', 5482, 1002);
```

```
INSERT INTO C_Order (Order_ID, C_Order, Price, Customer_ID)
```

```
VALUES (10002, 'Ruler-12, Pencil-20', 3678, 1001);
```

```
INSERT INTO C_Order (Order_ID, C_Order, Price, Customer_ID)
```

```
VALUES (10003, 'Facewash-12, Bodywash-20', 40566, 1003);
```

```
INSERT INTO C_Order (Order_ID, C_Order, Price, Customer_ID)
```

```
VALUES (10004, 'Facewash-8, Handwash-3', 10953, 1003);
```

```
INSERT INTO C_Order (Order_ID, C_Order, Price, Customer_ID)
```

```
VALUES (10005, 'CD-11, Hard_disc-9', 920547, 1005);
```

```
INSERT INTO c_reset_password (Customer_ID, Current_Password, New_Password)
```

```
VALUES (1001, 'Namal@1999', '1999@Namal');
```

```
INSERT INTO c_reset_password (Customer_ID, Current_Password, New_Password)
```

```
VALUES (1002, 'Kamal@1998', '1998@Namal');
```

```
INSERT INTO c_reset_password (Customer_ID, Current_Password, New_Password)
VALUES (1003, 'Shamali@2000', '2000@Shamali');
```

```
INSERT INTO c_reset_password (Customer_ID, Current_Password, New_Password)
VALUES (1004, 'Kamali@1998', '1998@Kamali');
```

```
INSERT INTO c_reset_password (Customer_ID, Current_Password, New_Password)
VALUES (1005, 'Pasindu1996', '1996@Pasindu');
```

```
INSERT INTO guest_user (Guest_User_ID, Guest_User_Name)
VALUES (01, 'Pamali');
```

```
INSERT INTO guest_user (Guest_User_ID, Guest_User_Name)
VALUES (02, 'Chehara');
```

```
INSERT INTO guest_user (Guest_User_ID, Guest_User_Name)
VALUES (03, 'Ravindu');
```

```
INSERT INTO guest_user (Guest_User_ID, Guest_User_Name)
VALUES (04, 'Takeshi');
```

```
INSERT INTO guest_user (Guest_User_ID, Guest_User_Name)
VALUES (05, 'Susiru');
```

```
INSERT INTO manage (Manage_ID, Name, Email)
VALUES (100001, 'Pavithra', 'pavithra@gmail.com');
```

```
INSERT INTO manage (Manage_ID, Name, Email)
VALUES (100002, 'Navindu', 'navindu@gmail.com');
```

```
INSERT INTO manage (Manage_ID, Name, Email)
VALUES (100003, 'Nimal', 'nimal@gmail.com');
```

```
INSERT INTO manage (Manage_ID, Name, Email)
VALUES (100004, 'Sachini', 'sachini@gmail.com');
```

```
INSERT INTO manage (Manage_ID, Name, Email)
VALUES (100005, 'Anjana', 'anjana@gmail.com');
```

```
INSERT INTO manage_contact_number (Manage_ID, Contact_Number)
VALUES (100001, 0786324512);
```

```
INSERT INTO manage_contact_number (Manage_ID, Contact_Number)
VALUES (100002, 0786624672);
```

```
INSERT INTO manage_contact_number (Manage_ID, Contact_Number)
VALUES (100002, 0716853546);
```

```
INSERT INTO manage_contact_number (Manage_ID, Contact_Number)
```

```
VALUES (100003, 0716365577);
```

```
INSERT INTO manage_contact_number (Manage_ID, Contact_Number)
```

```
VALUES (100004, 0716384561);
```

```
INSERT INTO manage_contact_number (Manage_ID, Contact_Number)
```

```
VALUES (100005, 0719782361);
```

```
INSERT INTO o_return (O_Return, Order_ID)
```

```
VALUES (10001/1, 10001);
```

```
INSERT INTO o_return (O_Return, Order_ID)
```

```
VALUES (10002/1, 10002);
```

```
INSERT INTO o_return (O_Return, Order_ID)
```

```
VALUES (10003/1, 10003);
```

```
INSERT INTO o_return (O_Return, Order_ID)
```

```
VALUES (10004/1, 10004);
```

```
INSERT INTO o_return (O_Return, Order_ID)
```

```
VALUES (10005/1, 10005);
```

```
INSERT INTO payment_option (Order_ID, Payment_Method)
```

```
VALUES (10001, 'Offline');
```

```
INSERT INTO payment_option (Order_ID, Payment_Method)
VALUES (10002, 'Online');
```

```
INSERT INTO payment_option (Order_ID, Payment_Method)
VALUES (10003, 'Online');
```

```
INSERT INTO payment_option (Order_ID, Payment_Method)
VALUES (10004, 'Offline');
```

```
INSERT INTO payment_option (Order_ID, Payment_Method)
VALUES (10005, 'Online');
```

```
INSERT INTO report (Report_Number, Title)
VALUES (200001, 'Salary');
```

```
INSERT INTO report (Report_Number, Title)
VALUES (200002, 'Order');
```

```
INSERT INTO report (Report_Number, Title)
VALUES (200003, 'Salary');
```

```
INSERT INTO report (Report_Number, Title)
VALUES (200004, 'Return');
```



```
INSERT INTO seller (Seller_ID, Name, Seller_Password, Email)
VALUES (501, 'Matheesha', 'Matheesha#501', 'matheesha@gmail.com');
```

```
INSERT INTO seller (Seller_ID, Name, Seller_Password, Email)
VALUES (502, 'Kasun', 'Kasun#502', 'kasun@gmail.com');
```

```
INSERT INTO seller (Seller_ID, Name, Seller_Password, Email)
VALUES (503, 'Nuwan', 'Nuwan#503', 'nuwan@gmail.com');
```

```
INSERT INTO seller (Seller_ID, Name, Seller_Password, Email)
VALUES (504, 'Namali', 'Namali#504', 'namali@gmail.com');
```

```
INSERT INTO seller (Seller_ID, Name, Seller_Password, Email)
VALUES (505, 'Bhashitha', 'Bhashitha#505', 'bhashitha@gmail.com');
```

```
INSERT INTO seller_contact_number (Seller_ID, Contact_Number)
VALUES (501, 0715632489);
```

```
INSERT INTO seller_contact_number(Seller_ID, Contact_Number)
VALUES (502, 0736452099);
```

```
INSERT INTO seller_contact_number (Seller_ID, Contact_Number)
VALUES (503, 0713642824);
```

```
INSERT INTO seller_contact_number (Seller_ID, Contact_Number)
```

```
VALUES (504, 0764852379);
```

```
INSERT INTO seller_contact_number (Seller_ID, Contact_Number)
```

```
VALUES (505, 0723679069);
```

```
INSERT INTO sellr_account (Seller_ID, Situation)
```

```
VALUES (501, 'Activated');
```

```
INSERT INTO sellr_account (Seller_ID, Situation)
```

```
VALUES (502, 'Deactivated');
```

```
INSERT INTO sellr_account (Seller_ID, Situation)
```

```
VALUES (503, 'Activated');
```

```
INSERT INTO sellr_account (Seller_ID, Situation)
```

```
VALUES (504, 'Activated');
```

```
INSERT INTO sellr_account (Seller_ID, Situation)
```

```
VALUES (505, 'Deactivated');
```

```
INSERT INTO social_media (Account_Name, A_Password, Type)
```

```
VALUES ('Namal1999', 'Ruwan$1999', 'Facebook');
```

```
INSERT INTO social_media (Account_Name, A_Password, Type)
```

```
VALUES ('Kamal1998', 'Kamal$1998', 'Instergam');
```

```
INSERT INTO social_media (Account_Name, A_Password, Type)
VALUES ('Shamali2000', 'Shamali$2000', 'Facebook');
```

```
INSERT INTO social_media (Account_Name, A_Password, Type)
VALUES ('Kamali1996', 'Kamali$1996', 'Imo');
```

```
INSERT INTO social_media (Account_Name, A_Password, Type)
VALUES ('Pasindu1996', 'Pasindu$1996', 'Imo');
```

```
INSERT INTO system_admin (Admin_ID, Name, Email)
VALUES ('300001', 'Nimashi', 'nimashi@gmail.com');
```

```
INSERT INTO system_admin (Admin_ID, Name, Email)
VALUES ('300002', 'Pavan', 'pavan@gmail.com');
```

```
INSERT INTO system_admin (Admin_ID, Name, Email)
VALUES ('300003', 'Vikum', 'vikum@gmail.com');
```

```
INSERT INTO system_admin (Admin_ID, Name, Email)
VALUES ('300004', 'Manudi', 'manudi@gmail.com');
```

```
INSERT INTO system_admin (Admin_ID, Name, Email)
VALUES ('300005', 'Praveen', 'praveen@gmail.com');
```

```
INSERT INTO s_reset_password (Seller_ID, Current_Password, Reset_Password)
VALUES (501, 'Matheesha#501', 'Matheesha@501');
```

```
INSERT INTO s_reset_password (Seller_ID, Current_Password, Reset_Password)
VALUES (502, 'Kasun#502', 'Kasun@502');
```

```
INSERT INTO s_reset_password (Seller_ID, Current_Password, Reset_Password)
VALUES (503, 'Nuwan#503', 'Nuwan@503');
```

```
INSERT INTO s_reset_password (Seller_ID, Current_Password, Reset_Password)
VALUES (504, 'Namali#504', 'Namali@504');
```

```
INSERT INTO s_reset_password (Seller_ID, Current_Password, Reset_Password)
VALUES (505, 'Bhashitha#505', 'Bhashitha@505');
```

```
INSERT INTO creat_report (Report_Number, Manage_ID)
VALUES (200001, 100002);
```

```
INSERT INTO creat_report (Report_Number, Manage_ID)
VALUES (200002, 100003);
```

```
INSERT INTO creat_report (Report_Number, Manage_ID)
VALUES (200003, 100001);
```

```
INSERT INTO creat_report (Report_Number, Manage_ID)
```

```
VALUES (200004, 100005);
```

```
INSERT INTO creat_report (Report_Number, Manage_ID)
```

```
VALUES (200005, 100001);
```

```
INSERT INTO admin_contact_number (Admin_ID, Contact_Number)
```

```
VALUES (300001, 0786245933);
```

```
INSERT INTO admin_contact_number (Admin_ID, Contact_Number)
```

```
VALUES (300002, 0713054853);
```

```
INSERT INTO admin_contact_number (Admin_ID, Contact_Number)
```

```
VALUES (300002, 0762485933);
```

```
INSERT INTO admin_contact_number (Admin_ID, Contact_Number)
```

```
VALUES (300002, 0710364503);
```

```
INSERT INTO admin_contact_number (Admin_ID, Contact_Number)
```

```
VALUES (300003, 0782368133);
```

```
INSERT INTO admin_contact_number (Admin_ID, Contact_Number)
```

```
VALUES (300004, 0786245933);
```

```
INSERT INTO admin_contact_number (Admin_ID, Contact_Number)
```

```
VALUES (300005, 0783657952);
```

INSERT INTO faq (Q_No, Question, Answer)

VALUES (500001, 'Why am I unable to complete my payment?', 'You might not be able to make your payment due to an issue associated with our system risk scanning or an issue faced with the bank related to your payment.

Exceeded card balance

Insufficient card or Dpoints funds

Blocked card

Expired card or Dpoints

Deactivated card');

INSERT INTO faq (Q_No, Question, Answer)

VALUES (500002, 'How can I tell whether a purchase includes installation for free?', 'Free installation is only available for specific goods. To learn more information about installation, make sure to read the product description.');

INSERT INTO faq (Q_No, Question, Answer)

VALUES (500003, "If I don't receive the OTP or verification code when I join up, what should I do?", "If you don't receive the OTP after signing up, please get in touch with us via live chat between the hours of 8 a.m. and 8 p.m. We will assist you as necessary.');

INSERT INTO faq (Q_No, Question, Answer)

VALUES (500004, 'What should I do if the post office refuses to accept my return package?', 'We make sure that all parcels with the postal return option are accepted by the post office. If there is a discrepancy, please contact customer care through live chat to report the issue.');

```
INSERT INTO faq (Q_No, Question, Answer)
```

```
VALUES (500005, 'Is using a credit or debit card secure?', 'We give fraud detection and prevention a high priority. We go out of our way to make sure that transactions are legitimate and that client information is completely safeguarded.');
```

```
INSERT INTO feedback (Customer_ID, Feedback, Rate)
```

```
VALUES (1001, 'Excellent', 10);
```

```
INSERT INTO feedback (Customer_ID, Feedback, Rate)
```

```
VALUES (1002, 'Good', 8);
```

```
INSERT INTO feedback (Customer_ID, Feedback, Rate)
```

```
VALUES (1003, 'Excellent', 10);
```

```
INSERT INTO feedback (Customer_ID, Feedback, Rate)
```

```
VALUES (1004, 'Neutral', 5);
```

```
INSERT INTO feedback (Customer_ID, Feedback, Rate)
```

```
VALUES (1005, 'Bad', 2);
```

7. Performance Requirement

To ensure excellent and positive customer interactions, a high-performing online customer support system must keep to a number of fundamental performance requirements. Some fundamental performance requirements for an online customer support system are listed below:

- **Response time:** The system needs to be responsive, providing clients as little waiting time as possible. In order to provide quick assistance and avoid client annoyance, the response time should be optimized. Within seconds or a few minutes, the system should acknowledge the request and offer an initial response.
- **Scalability:** The customer support system needs to be scalable in order to handle attribute levels of traffic and the customer questions. It should be able to handle higher user demands without changing in performance. This means that the system should be able to scale its resources to handle a higher number of interactions between customers.
- **Reliability:** Customer requests and interactions cannot be failed or discarded thanks to a responsible customer support system. To make sure that nothing gives through the cracks, it should have effective processes for maintaining and controlling the user request throughout the support process.
- **Availability:** Customers should be able to access and use the system whenever they need support. In order to reduce service failures and downtime, it should have a high uptime percentage. To maintain operation at all times, this may require employing redundant servers, load balancing, and failover methods
- **Multichannel Support:** Customers can choose for support through different methods, such as live chat, email, phone, or social media. Customers should be successful to select their preferred how of connection from any number of communication channels that the system has to manage effortlessly. Every channel should provide reliable and effective support.
- **Personalization:** A solid client service system should be able to customize communications. To offer personalized support, it must have access to customer profiles and applicable details. Details like customer history, preferences, earlier experiences, and purchase information may be included in this.
- **User-Friendly Interface:** Both customers and support personnel should be able to quickly access the customer support system. It should be simple to use, making it possible for customers to discover information fast and for agents to quickly access customer data. Features for simple but effective communication, such as chat windows or email templates, increase the user experience.
- **Performance Metrics and Analytics:** To track key performance indicators (KPIs), the system must provide complete performance metrics and analytics. This gives stats on agent productivity, workload distribution, customer satisfaction results, response and resolution times, and feedback from customers. These opinions help in improving the system and highlight up the possibility of improvements.

A successful and easy online customer support system will result from meeting these requirements for performance, improving customer satisfaction, loyalty, and overall support experiences.

9. Security Requirement

Security is a must for online customer support systems as they protect customer information and guarantee the system's confidentiality, integrity, and availability. The following are some basic safety standards to take into account:

- **Data Encryption:**

Implement strong encryption techniques to protect data transmission within the system as well as between the customer and support systems.

- **User Authentication:**

Include strong verification techniques in place to guarantee that only authorized users can access the customer support system. Depending on the importance of the data is, this can include with strong passwords or multi-factor authentication.

- **Access Controls:**

To limit system access based on user roles and privileges, apply access control techniques. Limit authorized support staff's access to client data.

- **Secure Storage:**

Make sure that access controls and appropriate encryption techniques are used to store customer data securely. Implement security controls like firewalls, intrusion detection systems, and continual security audits to prevent unwanted access.

- **Regular Updates and Patching:**

Maintain the latest security fixes and updates for the customer support system's underlying software and infrastructure to address any known vulnerabilities.

- **Data Backup and Recovery:**

Create reliable data recovery procedures and commonly back up client data for safe company continuity in situation of a system failure or data loss.