Sri Lanka Institute of Information Technology

Final Group Project.

MLB_16.01_08

Online Auction System

IT1090 – Information Systems and Data Modeling (ISDM)

B.Sc. (Hons) in Information Technology

Group Details:

	Student Id	Student Name	Email	Contact Number
1	IT22564740	Heshan Y.B.K	It22564740@my.sliit.lk	0703767354
2	IT22138354	Aloka M.D.C	It22138354@my.sliit.lk	0719404928
3	IT22620248	Sandakalum A.R.G.S.S	it22620248@my.sliit.lk	0760075198
4	IT22600998	Bandara R.M.S.C	It22600998@my.sliit.lk	0718764555
5	IT22550644	Samarasinghe C.K	It22550644@my.sliit.lk	0703702890

Table of content.

Introduction

An online auction system is a platform where individuals or businesses can buy and sell goods or services online. It eliminates the need for physical auctions and allows participants to engage in bidding remotely. Users can browse through various listings, bid on items they are interested in, and track auction progress in real-time.

Hypothetical scenario

This online auction system is divided into three segments. Those are buying, selling, shipping. These can be easily redeemed after accessing our platform. Firstly, you go to the website and create the account in it. Next step is log in the account using the password and username from system. After it can be watched. Product buying and selling are the main activities in this system. In addition to that shipping is another necessary activity for it.

After the create account, can be sign in. then after can be select what is want you? Like sell or buy. Can be selected once. Next step is “Next” button can be select. We think we selected “buy” for purchase item. After can be search specific item. Can be click on specific items. After can be enter the information and pay money. Next email and SMS are received by buyer. After, buyer verification it. Can be enter the shipping details.

Lastly, user can be providing feedback about over system.

Requirements Analysis

Main Requirements

Main requirements typically outline the core capabilities or key components of the system. The following requirements are the main requirements for our online auction system website.

1. User Registration – Users should be able to provide the necessary information to create accounts and participate in auctions.
2. Auction Listing - On the seller's end, it should be possible to list items for auction, including details such as item name, description, starting price, etc.
3. Bidding – On the buyer ' s end, should be able to place bids for their interested items.
4. Auction Management – The website should have a mechanism for managing ongoing auctions, including bid tracking, display remaining time, etc.
5. Payment Processing - To process transactions between buyers and sellers, the system must provide payment gateways.
6. Notifications – The system must notify the users regarding the current status of their auctions.
7. Feedback and Ratings – Buyers should be able to leave feedback and rating for auction items based on their auction experience.
8. Contact - Buyers should be able to obtain seller contact details and both of buyer and seller able to obtain the contact details of system administration.

Functional Requirements

Functional requirements are what users expect from the system. Our online auction system has several functional requirements, and these requirements are based on the type of user.

1. Guest (Unregistered user) requirements

- ✓ Guests only can access the home page, Contact page, Auction list page Login and Sign-up page without the ability to place bids and participating in auctions.
- ✓ These types of users can only view the auction and item details.
- ✓ Guest users should be able to read customers' feedback and rating on their preference items.
- ✓ Guest user should be able to create account with providing necessary and accurate details about him.

2. Buyer

- ✓ Buyers should have the ability to place bids on auctioned items.
- ✓ Buyers should be able to add auctions to their watchlist to track items of interest.
- ✓ Buyers must receive the notification regarding the auction results and payments reminders.
- ✓ Buyers should be able to view their bidding history and participated auctions details.
- ✓ Buyers should be able to provide feedback and ratings to items and sellers.
- ✓ System should provide facility for buyers to chat with sellers and other participants during the auction.

3. Seller

- ✓ The seller should have the ability to list the items for auctions.
- ✓ Sellers must be notified when the auction of their items begins, an item has been sold and the payment for it has been completed.
- ✓ Sellers should be able to view the feedback on their items.
- ✓ System should provide the facility for sellers to chat with buyers during the auction time.

4. Administrator

- ✓ The administrator should have the authority of managing accounts including the activities of account creation, suspension, verification and deletion.
- ✓ The administrator should be able to manage all auctions.
- ✓ Administrators should have the ability to configure system settings.
- ✓ The administrator should be able to obtain an analysis report regarding the auctions conducted throughout the day.

Non-Functional Requirements

Non-functional requirements define the properties and characteristics of a system beyond its specific functionality. Non-functional requirements may be more critical than functional requirements and without these the system can be useless.

Usability

- ✓ The system should have an efficient interface.
- ✓ The system should be easy for users to navigate, search and bid for items.
- ✓ The system must be user-friendly.

Security

- ✓ Transactions in the system must be very secure.
- ✓ User details must be kept confidential.
- ✓ System should be able to prevent unauthorized access.

Availability

- ✓ The website should be available 24 hours.

Speed

- ✓ Web pages must be loaded into the browser in a few seconds.
- ✓ The system must provide solutions for user complaints in one hour.

Flexibility

- ✓ The system should be flexible with advanced settings.

Data Requirements

For Buyer

- Buyer_id,
- First_Name
- Last_Name
- DoB
- Home_Address
- Country
- Postal_Code
- Email
- Phone_No,
- Buyer_UserName,
- Buyer_Password

For Seller

- Seller_id
- First_Name
- Last_Name
- DoB
- Home_Address,
- Country
- Postal_Code
- Email
- Phone_No
- Seller_UserName

- Seller_Password

For Shipment Manager

- Manager ID
- Full Name
- Username
- Password
- Contact Number
- Email-Address
- Date of Birth

For Item

- Item ID
- Item Name
- Seller ID
- Category
- Description
- Starting Price
- Customer Feedbacks

For Placing a Bid

- Bid ID
- Item ID
- Item Name

- Duration
- Bidding Amount
- Bidder Place

For Won Item

- Item ID
- Item Name
- Bidding Amount
- Buyer ID
- Seller ID

For Payments

- Payment ID
- Date & Time
- Item ID
- Item Name
- Buyer ID
- Seller ID
- Amount

For Shipment

- Shipment ID
- Item ID

- Buyer ID
- Shipment-Address
- Shipment Status

For Feedback

- Feedback ID
- Buyer ID
- Item Name
- Item Rating
- Comment

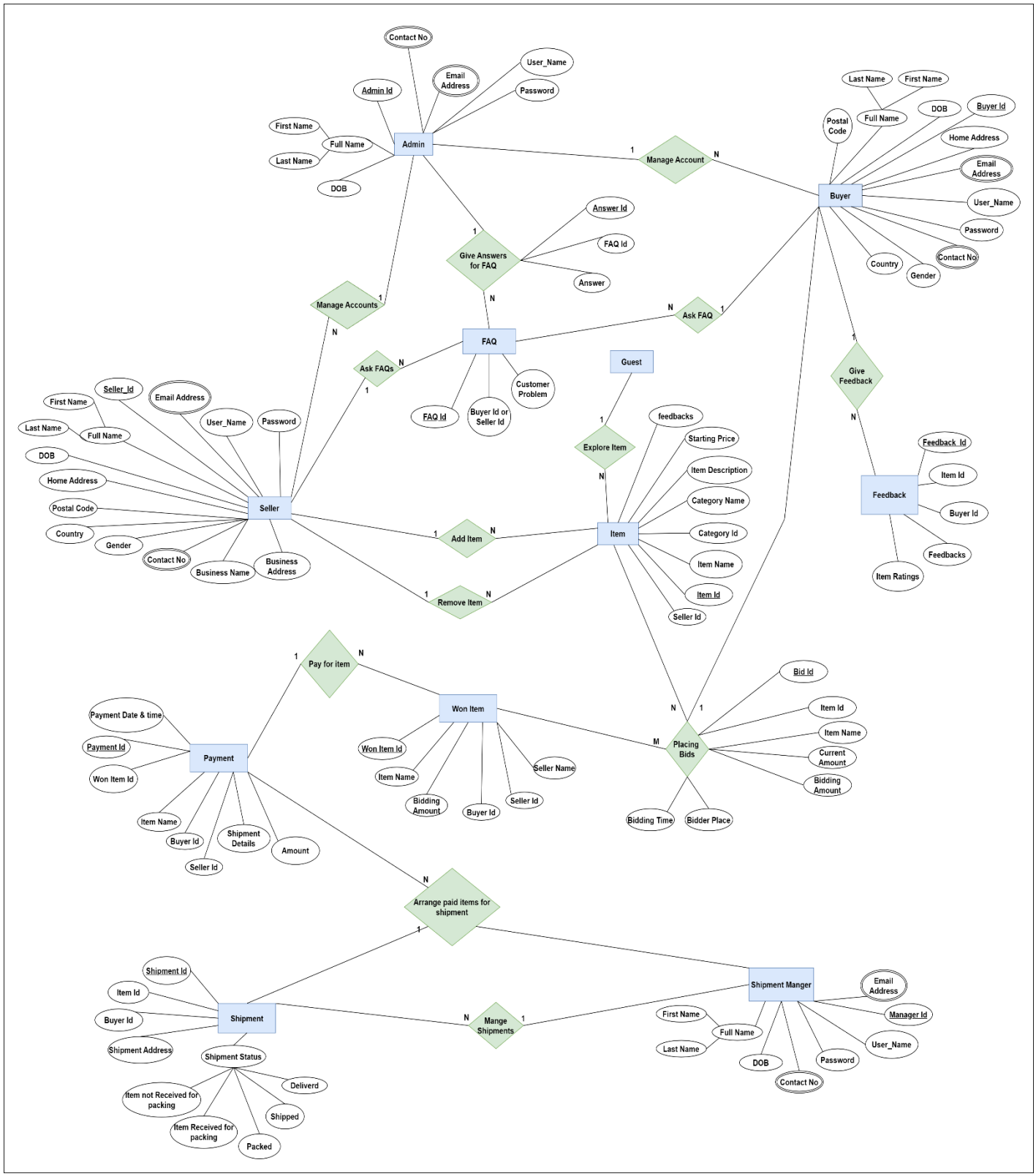
For FAQ

- FAQ ID
- Buyer Or Seller ID
- Customer Problem

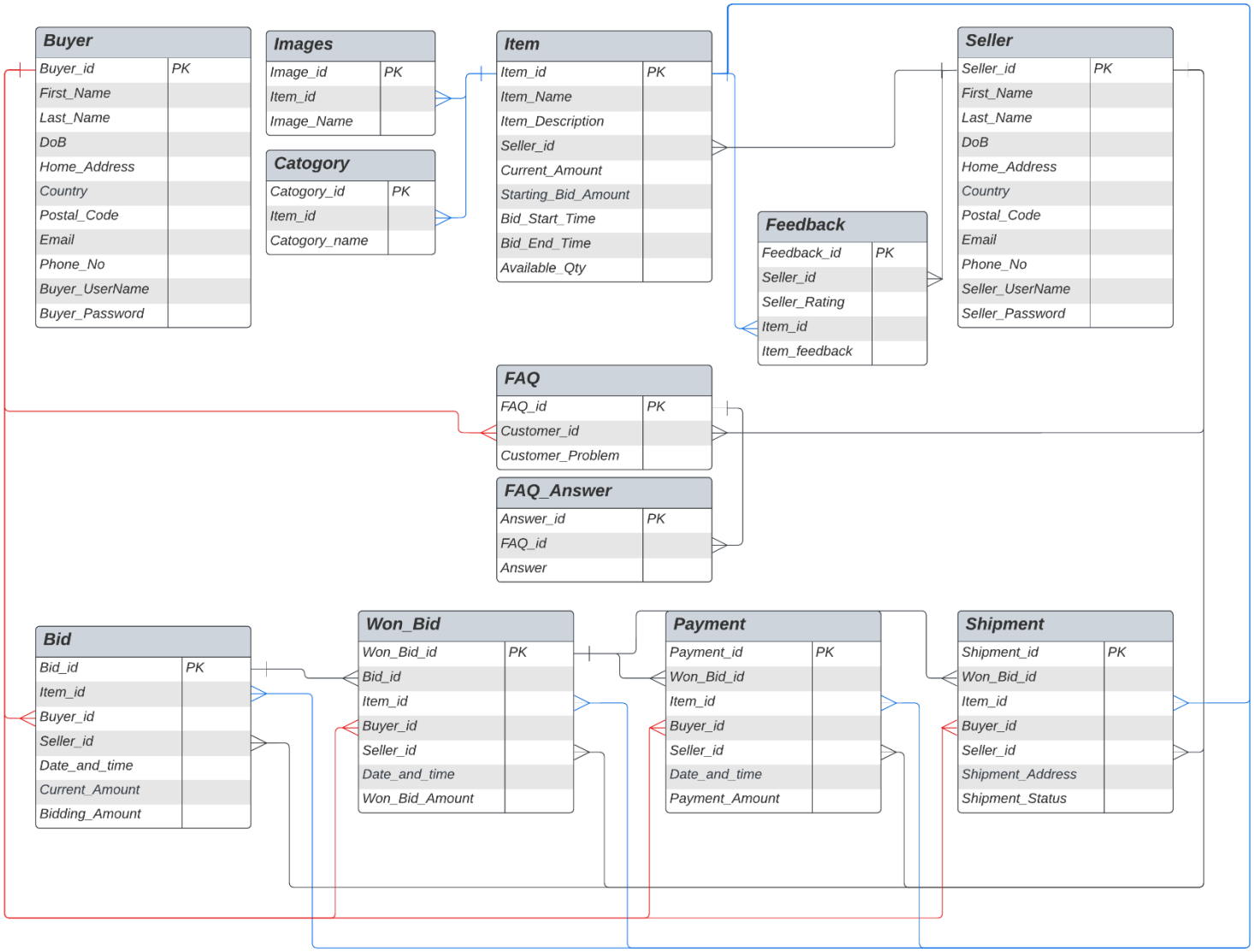
Answer For FAQ

- Answer ID
- FAQ ID
- Answer

Entity Relationship (E-R) Diagram



Relational Schemas



SQL Queries

```
create database auction_sys


-- Buyer Table --

create table Buyer(

Buyer_id varchar(10)not null,

First_Name varchar(200)not null,

Last_Name varchar(200),

DoB date not null,

Home_Address varchar(200)not null,

Country varchar(50)not null,

Postal_Code int,

Email varchar(50)not null,

Phone_No varchar(20)not null,

Buyer_UserName varchar(20)not null,

Buyer_Password varchar(20)not null,


CONSTRAINT pk_buy PRIMARY KEY(Buyer_id),

);


-- Insert values into Buyer Table --

INSERT INTO Buyer (Buyer_id, First_Name, Last_Name, DoB, Home_Address, Country, Postal_Code,
Email, Phone_No, Buyer_UserName, Buyer_Password)

VALUES

  ('B00001', 'John', 'Doe', '1985-05-15', '123 Main St', 'United States', 12345, 'john.doe@example.com',
'555-123-4567', 'johndoe', 'password123'),

  ('B00002', 'Jane', 'Smith', '1990-09-22', '456 Elm St', 'Canada', 67890, 'jane.smith@example.com', '555-
987-6543', 'janesmith', 'securepass'),

  ('B00003', 'David', 'Johnson', '1978-03-10', '789 Oak Ave', 'United Kingdom', 56789,
'david.johnson@example.com', '555-456-7890', 'davidj', 'pass123'),
```

('B00004', 'Sarah', 'Davis', '1995-07-28', '321 Maple Rd', 'Australia', 54321, 'sarah.davis@example.com', '555-789-0123', 'sarahd', 'p@$$w0rd'),

('B00005', 'Michael', 'Wilson', '1982-01-05', '567 Pine Ln', 'Germany', 34567, 'michael.wilson@example.com', '555-012-3456', 'michaelw', 'mypass123'),

('B00006', 'Emily', 'Anderson', '1993-11-12', '890 Cedar St', 'France', 45678, 'emily.anderson@example.com', '555-234-5678', 'emilya', 'password321');

-- Seller Table --

create table Seller(

Seller_id varchar(10)not null,

First_Name varchar(200)not null,

Last_Name varchar(200),

DoB date not null,

Home_Address varchar(200)not null,

Country varchar(50)not null,

Postal_Code int,

Email varchar(50)not null,

Phone_No varchar(20)not null,

Seller_UserName varchar(20)not null,

Seller_Password varchar(20)not null,

CONSTRAINT pk_sell PRIMARY KEY(Seller_id),

);

-- Insert values into Seller Table --

INSERT INTO Seller (Seller_id, First_Name, Last_Name, DoB, Home_Address, Country, Postal_Code, Email, Phone_No, Seller_UserName, Seller_Password)

VALUES

('S00001', 'Mark', 'Johnson', '1980-08-12', '123 Main St', 'United States', 12345, 'mark.johnson@example.com', '555-123-4567', 'markj', 'password123'),

('S00002', 'Emily', 'Smith', '1992-04-17', '456 Elm St', 'Canada', 67890, 'emily.smith@example.com', '555-987-6543', 'emilys', 'securepass'),

('S00003', 'Alex', 'Davis', '1975-12-25', '789 Oak Ave', 'United Kingdom', 56789, 'alex.davis@example.com', '555-456-7890', 'alexd', 'pass123'),

('S00004', 'Sophia', 'Wilson', '1988-06-30', '321 Maple Rd', 'Australia', 54321, 'sophia.wilson@example.com', '555-789-0123', 'sophiaw', 'mypass123'),

('S00005', 'Jacob', 'Brown', '1993-02-14', '567 Pine Ln', 'Germany', 34567, 'jacob.brown@example.com', '555-012-3456', 'jacobb', 'password321'),

('S00006', 'Olivia', 'Anderson', '1987-10-10', '890 Cedar St', 'France', 45678, 'olivia.anderson@example.com', '555-234-5678', 'oliviaa', 'securepassword');

-- Item Table --

create table Item (

Item_id varchar(10)not null,

Item_Name varchar(20)not null,

Item_Description varchar(200),

Seller_id varchar(10),

Current_Amount varchar(20),

Starting_Bid_Amount varchar(20),

Bid_Start_Time TIMESTAMP,

Bid_End_Time TIMESTAMP,

Available_Qty int,

CONSTRAINT pk_item PRIMARY KEY(Item_id),

CONSTRAINT fk_sell FOREIGN KEY(Seller_id) REFERENCES Seller(Seller_id),

);

-- Insert values into Item Table --

INSERT INTO Item (Item_id, Item_Name, Item_Description, Seller_id, Current_Amount, Starting_Bid_Amount, Bid_Start_Time, Bid_End_Time, Available_Qty)

VALUES

('I00001', 'Stainless Watch', 'Stainless steel wristwatch with leather strap', 'S00001', '$50.00', '$30.00', '2023-06-04 10:00:00', '2023-06-08 18:00:00', 1),

('I00002', 'Digital Camera', 'Digital camera with 20MP resolution', 'S00002', '$200.00', '$150.00', '2023-06-05 12:00:00', '2023-06-09 20:00:00', 2),

('I00003', '15.6" Laptop', '15.6-inch laptop with Intel Core i5 processor', 'S00003', '$800.00', '$700.00', '2023-06-06 14:00:00', '2023-06-10 22:00:00', 1),

('I00004', 'Wireless Headphones', 'Wireless Bluetooth headphones with noise cancellation', 'S00004', '$100.00', '$80.00', '2023-06-07 16:00:00', '2023-06-11 00:00:00', 3),

('I00005', 'Gaming Console', 'Xbox Series X gaming console with 1TB storage', 'S00005', '$500.00', '$400.00', '2023-06-08 18:00:00', '2023-06-12 04:00:00', 2),

('I00006', 'iPhone 13 Pro', 'Apple iPhone 13 Pro with 256GB storage', 'S00006', '$1,200.00', '$1,000.00', '2023-06-09 20:00:00', '2023-06-13 08:00:00', 1);

-- Images Table --

CREATE TABLE Images (

Image_id varchar(10)not null,

Item_id varchar(10)not null,

Image_Name varchar(100),

CONSTRAINT pk_image PRIMARY KEY(Image_id),

CONSTRAINT fk_itme FOREIGN KEY(Item_id) REFERENCES Item(Item_id),

);

-- Insert values into Images Table --

INSERT INTO Images (Image_id, Item_id, Image_Name)

VALUES

('IMG00001', 'I00001', 'watch_front.jpg'),

('IMG00002', 'I00001', 'watch_back.jpg'),

('IMG00003', 'I00002', 'camera_front.jpg'),

('IMG00004', 'I00002', 'camera_side.jpg'),

('IMG00005', 'I00003', 'laptop_open.jpg'),

('IMG00006', 'I00003', 'laptop_closed.jpg');

--Catagory Table--

```
CREATE TABLE Catogory(
Catogory_id varchar(10)not null,
Item_id varchar(10)not null,
Catogory_name varchar(50)not null,

CONSTRAINT pk_catogory PRIMARY KEY(Catogory_id),
CONSTRAINT fk_itme FOREIGN KEY(Item_id) REFERENCES Item(Item_id)
);

-- Insert values into Catagory Table --
INSERT INTO Category (Category_id, Item_id, Category_name)
VALUES
 ('C00001', 'I00001', 'Watches'),
 ('C00002', 'I00002', 'Cameras'),
 ('C00003', 'I00003', 'Laptops'),
 ('C00004', 'I00004', 'Headphones'),
 ('C00005', 'I00005', 'Gaming Consoles'),
 ('C00006', 'I00006', 'Smartphones');

-- FAQ Table --
create table FAQ (
FAQ_id varchar(10)not null,
Customer_id varchar(10)not null,
Customer_Problem varchar(200),

CONSTRAINT pk_faq PRIMARY KEY(FAQ_ID),
CONSTRAINT fk_customer1 FOREIGN KEY(Customer_ID) REFERENCES Buyer(Buyer_id),
CONSTRAINT fk_customer2 FOREIGN KEY(Customer_ID) REFERENCES Seller(Seller_id),
);
```

```
-- Insert values into FAQ Table --
INSERT INTO FAQ (FAQ_id, Customer_id, Customer_Problem)
VALUES
  ('FAQ00001', 'C00001', 'I have an issue with my order delivery.'),
  ('FAQ00002', 'C00002', 'How can I track my package?'),
  ('FAQ00003', 'C00003', 'I received a damaged item. What should I do?'),
  ('FAQ00004', 'C00004', 'What is your return policy?'),
  ('FAQ00005', 'C00005', 'How can I change my shipping address?'),
  ('FAQ00006', 'C00006', 'Do you offer international shipping?');

-- FAQ_Answer Table --
create table FAQ_Answer (
Answer_id varchar(10)not null,
FAQ_id varchar(10)not null,
Answer varchar(200),

CONSTRAINT pk_answer PRIMARY KEY(Answer_id),
CONSTRAINT fk_faq FOREIGN KEY(FAQ_id) REFERENCES FAQ(FAQ_id),
);

-- Insert values into FAQ_Answer Table --
INSERT INTO FAQ_Answer (Answer_id, FAQ_id, Answer)
VALUES
  ('A00001', 'FAQ00001', 'Please contact our customer support team for assistance with your order delivery.'),
  ('A00002', 'FAQ00002', 'You can track your package by logging into your account and accessing the order tracking section.'),
  ('A00003', 'FAQ00003', 'We apologize for the inconvenience. Please reach out to our customer support team with details and we will assist you.'),
  ('A00004', 'FAQ00004', 'Our return policy allows for returns within 30 days of purchase. Please refer to our website for detailed instructions.'),
```

('A00005', 'FAQ00005', 'To change your shipping address, go to your account settings and update your address information.'),

('A00006', 'FAQ00006', 'Yes, we offer international shipping to select countries. Please check our website for the list of eligible destinations.');

-- Feedback Table --

```
create table Feedback (

Feedback_id varchar(10)not null,

Seller_id varchar(10)not null,

Seller_Rating int,

Item_id varchar(10)not null,

Item_feedback varchar(200),


CONSTRAINT pk_feedbk PRIMARY KEY(Feedback_id),

CONSTRAINT fk_sell FOREIGN KEY(Seller_id) REFERENCES Seller(Seller_id),

CONSTRAINT fk_item FOREIGN KEY(Item_id) REFERENCES Item(Item_id),

);
```

-- Insert values into Feedback Table --

```
INSERT INTO Feedback (Feedback_id, Seller_id, Seller_Rating, Item_id, Item_feedback)

VALUES

('F00001', 'S00001', 4, 'I00001', 'Great seller! The item arrived on time and in excellent condition.'),

('F00002', 'S00002', 5, 'I00002', 'The camera I bought is amazing! It exceeded my expectations.'),

('F00003', 'S00003', 3, 'I00003', 'The laptop works fine, but there were some scratches on the lid.'),

('F00004', 'S00004', 5, 'I00004', 'The headphones have great sound quality and are very comfortable to wear.'),

('F00005', 'S00005', 2, 'I00005', 'I had a lot of trouble setting up the gaming console. The instructions were not clear.'),

('F00006', 'S00006', 4, 'I00006', 'The iPhone is fantastic! Its fast and the camera takes stunning photos.');
```

-- Bid table --

```
CREATE TABLE Bid(
```

```
Bid_id varchar(10)not null,
Item_id varchar(10)not null,
Buyer_id varchar(10)not null,
Seller_id varchar(10)not null,
Date_and_time TIMESTAMP,
Current_Amount int,
Bidding_Amount int,

CONSTRAINT pk_bid PRIMARY KEY(Bid_id),
CONSTRAINT fk_item FOREIGN KEY(Item_id) REFERENCES Item(Item_id),
CONSTRAINT fk_buy FOREIGN KEY(Buyer_id) REFERENCES Buyer(Buyer_id),
CONSTRAINT fk_sell FOREIGN KEY(Seller_id) REFERENCES Seller(Seller_id)
);

-- Insert values into Bid Table --
INSERT INTO Bid (Bid_id, Item_id, Buyer_id, Seller_id, Date_and_time, Current_Amount,
Bidding_Amount)
VALUES
  ('B00001', 'I00001', 'B00001', 'S00001', '2023-06-04 10:15:00', 50, 60),
  ('B00002', 'I00001', 'B00002', 'S00001', '2023-06-04 10:30:00', 60, 70),
  ('B00003', 'I00002', 'B00003', 'S00002', '2023-06-04 11:00:00', 200, 250),
  ('B00004', 'I00002', 'B00004', 'S00002', '2023-06-04 11:15:00', 250, 300),
  ('B00005', 'I00003', 'B00005', 'S00003', '2023-06-04 11:30:00', 800, 900),
  ('B00006', 'I00003', 'B00006', 'S00003', '2023-06-04 11:45:00', 900, 1000);

-- Won bid table --
CREATE TABLE Won_Bid(
Won_Bid_id varchar(10)not null,
Bid_id varchar(10)not null,
Item_id varchar(10)not null,
Buyer_id varchar(10)not null,
```

```
Seller_id varchar(10)not null,
Date_and_time TIMESTAMP,
Won_Bid_Amount int,

CONSTRAINT pk_wonbid PRIMARY KEY(Won_Bid_id),
CONSTRAINT fk_bid FOREIGN KEY(Bid_id) REFERENCES Bid(Bid_id),
CONSTRAINT fk_item FOREIGN KEY(Item_id) REFERENCES Item(Item_id),
CONSTRAINT fk_buy FOREIGN KEY(Buyer_id) REFERENCES Buyer(Buyer_id),
CONSTRAINT fk_sell FOREIGN KEY(Seller_id) REFERENCES Seller(Seller_id)
);

INSERT INTO Won_Bid (Won_Bid_id, Bid_id, Item_id, Buyer_id, Seller_id, Date_and_time,
Won_Bid_Amount)
VALUES
 ('W00001', 'B00002', 'I00001', 'B00002', 'S00001', '2023-06-04 10:35:00', 70),
 ('W00002', 'B00004', 'I00002', 'B00004', 'S00002', '2023-06-04 11:20:00', 300),
 ('W00003', 'B00006', 'I00003', 'B00006', 'S00003', '2023-06-04 11:50:00', 1000);

-- Payment Table --
CREATE TABLE Payment(
Payment_id varchar(10)not null,
Won_Bid_id varchar(10)not null,
Item_id varchar(10)not null,
Buyer_id varchar(10)not null,
Seller_id varchar(10)not null,
Date_and_time TIMESTAMP,
Payment_Amount int,

CONSTRAINT pk_pay PRIMARY KEY(Payment_id),
CONSTRAINT fk_bid FOREIGN KEY(Won_Bid_id) REFERENCES Won_Bid(Won_Bid_id),
CONSTRAINT fk_item FOREIGN KEY(Item_id) REFERENCES Item(Item_id),
```

```
CONSTRAINT fk_buy FOREIGN KEY(Buyer_id) REFERENCES Buyer(Buyer_id),
CONSTRAINT fk_sell FOREIGN KEY(Seller_id) REFERENCES Seller(Seller_id),
);

INSERT INTO Payment (Payment_id, Won_Bid_id, Item_id, Buyer_id, Seller_id, Date_and_time,
Payment_Amount)
VALUES
  ('P00001', 'W00001', 'I00001', 'B00002', 'S00001', '2023-06-05 12:00:00', 70),
  ('P00002', 'W00002', 'I00002', 'B00004', 'S00002', '2023-06-06 14:30:00', 300),
  ('P00003', 'W00003', 'I00003', 'B00006', 'S00003', '2023-06-07 10:15:00', 1000);

-- Shipment Table --
CREATE TABLE Shipment(
Shipment_id varchar(10)not null,
Won_Bid_id varchar(10)not null,
Item_id varchar(10)not null,
Buyer_id varchar(10)not null,
Seller_id varchar(10)not null,
Shipment_Address varchar(200),
Shipment_Status varchar(20),

CONSTRAINT pk_shipment PRIMARY KEY(Shipment_id),
CONSTRAINT fk_bid FOREIGN KEY(Won_Bid_id) REFERENCES Won_Bid(Won_Bid_id),
CONSTRAINT fk_item FOREIGN KEY(Item_id) REFERENCES Item(Item_id),
CONSTRAINT fk_buyer FOREIGN KEY(Buyer_id) REFERENCES Buyer(Buyer_id),
CONSTRAINT fk_seller FOREIGN KEY(Seller_id) REFERENCES Seller(Seller_id),
);
```

```
INSERT INTO Shipment (Shipment_id, Won_Bid_id, Item_id, Buyer_id, Seller_id, Shipment_Address, Shipment_Status)
VALUES
  ('S00001', 'W00001', 'I00001', 'B00002', 'S00001', '123 Main St, Cityville', 'In Transit'),
  ('S00002', 'W00002', 'I00002', 'B00004', 'S00002', '456 Elm St, Townsville', 'Delivered'),
  ('S00003', 'W00003', 'I00003', 'B00006', 'S00003', '789 Oak St, Villageton', 'Out for Delivery');
```

Performance Considerations

- Our online auction system is designed to accommodate the simultaneous usage of more than 100 users, ensuring a smooth and efficient auction experience for all participants.
- Users can access the system anytime, from any device, providing flexibility and convenience.
- The login process is optimized to verify user credentials within a few seconds, allowing users to quickly access the auction platform.
- Bid submissions and updates are processed instantly, with each request completed within 10 seconds.
- Upon successful registration, users receive a verification link via email to confirm their account.
- For each auction item, the system automatically sends email notifications to the winning bidder, providing them with a payment invoice and instructions for completing the transaction.
- System updates and database optimizations are completed within 2 seconds, ensuring minimal disruption to ongoing auctions, and maintaining data integrity.
- Search functionality within the system allows users to quickly find specific auction items or categories of interest, ensuring a seamless browsing experience.
- Our auction system accepts various payment methods, providing users with flexibility while ensuring the protection of their financial information.
- The user interface of the auction system is carefully designed to be unique and user-friendly, enabling participants to navigate the platform effortlessly and engage in bidding activities with ease.

Security Requirements

- During registration, users must confirm all the information they provide, ensuring its accuracy and completeness.

- Only registered users are granted access to login to the system, providing a secure environment for authorized participants.

- During the registration process, users are required to create a strong password that includes a combination of uppercase and lowercase letters, numbers, and symbols, enhancing the security of their accounts.

- The admin dashboard or admin's page is restricted from general access.

- Our system uses one-time password method for all online transactions to ensure secure payments

- All payments made through the system are validated by the relevant banking institutions.

- The privacy of customer inquiries and questions is protected.

- Only administrators have the authority to reply to feedback provided by users, while other customers can read and benefit from the feedback.

- Administrators have the capability to modify system data as necessary, ensuring the system remains up to date.

- Regular database backups are conducted to ensure the safety and integrity of all system data, allowing for recovery in the event of any unforeseen issues or data loss.

- The online auction system is equipped with robust security measures to protect against viruses, malware, and other malicious threats.