

# Logical Data Model Report

## Task 1: Entity-Relationship (ER) Model

### Entities and Attributes with Data Types:

1. **Client (User):**
  - UserID (PK, INT, NOT NULL)
  - Name (VARCHAR(100), NOT NULL)
  - Email (VARCHAR(100), UNIQUE, NOT NULL)
  - PhoneNumber (VARCHAR(15), UNIQUE, NOT NULL)
  - BiometricData (VARBINARY, NOT NULL)
  - AccountID (FK, INT, NOT NULL)
2. **Account:**
  - AccountID (PK, INT, NOT NULL)
  - UserID (FK, INT, NOT NULL)
  - AccountNumber (VARCHAR(20), UNIQUE, NOT NULL)
  - AccountBalance (DECIMAL(10, 2), DEFAULT 0.00)
  - DateCreated (DATETIME, NOT NULL)
3. **Merchant:**
  - MerchantID (PK, INT, NOT NULL)
  - MerchantName (VARCHAR(100), NOT NULL)
  - Location (VARCHAR(255), NOT NULL)
4. **Merchant\_Client:**
  - MerchantClientID (PK, INT, NOT NULL)
  - MerchantID (FK, INT, NOT NULL)
  - UserID (FK, INT, NOT NULL)
  - InteractionDate (DATETIME, NOT NULL)
  - InteractionType (VARCHAR(50), NOT NULL)
5. **Item:**
  - ItemID (PK, INT, NOT NULL)
  - ItemName (VARCHAR(100), NOT NULL)
  - ItemPrice (DECIMAL(10, 2), NOT NULL)
6. **Transaction\_Items:**
  - TransactionItemID (PK, INT, NOT NULL)
  - TransactionID (FK, INT, NOT NULL)
  - ItemID (FK, INT, NOT NULL)
  - Quantity (INT, NOT NULL)
  - Price (DECIMAL(10, 2), NOT NULL)
7. **Scanner:**
  - ScannerID (PK, INT, NOT NULL)
  - Location (VARCHAR(100), NOT NULL)
  - ScannerType (VARCHAR(50), NOT NULL)
  - UserID (FK, INT, NOT NULL)
8. **Authentication Server:**
  - AuthID (PK, INT, NOT NULL)
  - UserID (FK, INT, NOT NULL)
  - ScannerID (FK, INT, NOT NULL)
  - BiometricData (VARBINARY, NOT NULL)

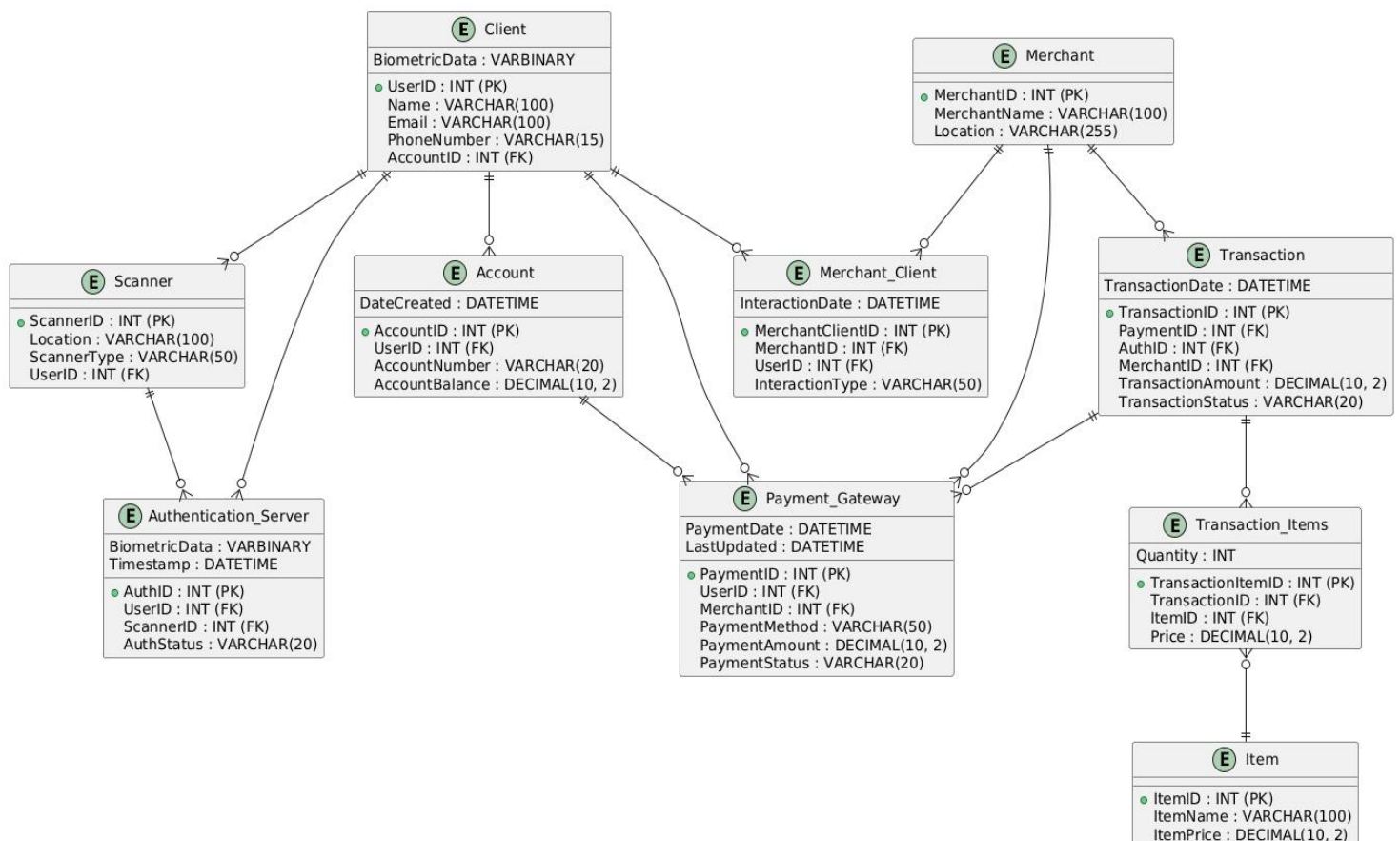
- AuthStatus (VARCHAR(20), NOT NULL)
- Timestamp (DATETIME, NOT NULL)

#### 9. Payment Gateway:

- PaymentID (PK, INT, NOT NULL)
- UserID (FK, INT, NOT NULL)
- MerchantID (FK, INT, NOT NULL)
- PaymentMethod (VARCHAR(50), NOT NULL)
- PaymentAmount (DECIMAL(10, 2), NOT NULL)
- PaymentStatus (VARCHAR(20), NOT NULL)
- PaymentDate (DATETIME, NOT NULL)
- LastUpdated (DATETIME, NOT NULL)

#### 10. Transaction:

- TransactionID (PK, INT, NOT NULL)
- PaymentID (FK, INT, NOT NULL)
- AuthID (FK, INT, NOT NULL)
- MerchantID (FK, INT, NOT NULL)
- TransactionAmount (DECIMAL(10, 2), NOT NULL)
- TransactionStatus (VARCHAR(20), NOT NULL)
- TransactionDate (DATETIME, NOT NULL)



**Entity-Relationship (ER) Model Diagram**

## Task 2: Relationships & Constraints

1. **Client - Account:** One-to-One
2. **Client - Authentication Server:** One-to-One
3. **Client - Merchant\_Client:** Many-to-Many
4. **Authentication Server - Payment Gateway:** One-to-One
5. **Payment Gateway - Transaction:** One-to-One
6. **Transaction - Merchant:** One-to-One
7. **Client - Scanner:** One-to-Many
8. **Transaction - Transaction\_Items:** One-to-Many
9. **Item - Transaction\_Items:** One-to-Many

### Constraints:

- **NOT NULL:** Applied to all essential fields.
- **UNIQUE:** Applied to Email, PhoneNumber, AccountNumber.
- **CHECK:** PaymentAmount, AccountBalance  $\geq 0.00$ .
- **DEFAULT:** AccountBalance = 0.00.
- **Foreign Key Constraints:** Enforce data integrity across tables.
- **Data Encryption:** Applied to BiometricData.

## Task 3: Normalization (3rd Normal Form)

- All tables are now in 3NF.
- `Transaction_Items` resolves the many-to-many relationship between Transactions and Items.
- `Merchant_Client` resolves the many-to-many relationship between Merchants and Clients.
- Account and Item data are now managed independently, reducing redundancy.

## Task 4: Handling Data Scenarios

### Scenario 1: New User Registration

- Client registers, Account created, AccountBalance defaults to 0.00.

### Scenario 2: Multiple Item Purchase in a Single Transaction

- Client selects multiple items, a single transaction is initiated.
- Each item is recorded in `Transaction_Items` with its quantity and price.

### Scenario 3: Refund or Chargeback Handling

- If a refund occurs, `TransactionStatus` is updated to 'Refunded'.
- Refund details are recorded in `Transaction_Items` with negative quantities.

#### **Scenario 4: Data Consistency Check**

- Orphaned records in `Transaction_Items` and `Merchant_Client` are validated through FK constraints.

#### **Scenario 5: Data Security and Access Control**

- `BiometricData` is encrypted.
- Sensitive data access is restricted based on user roles.