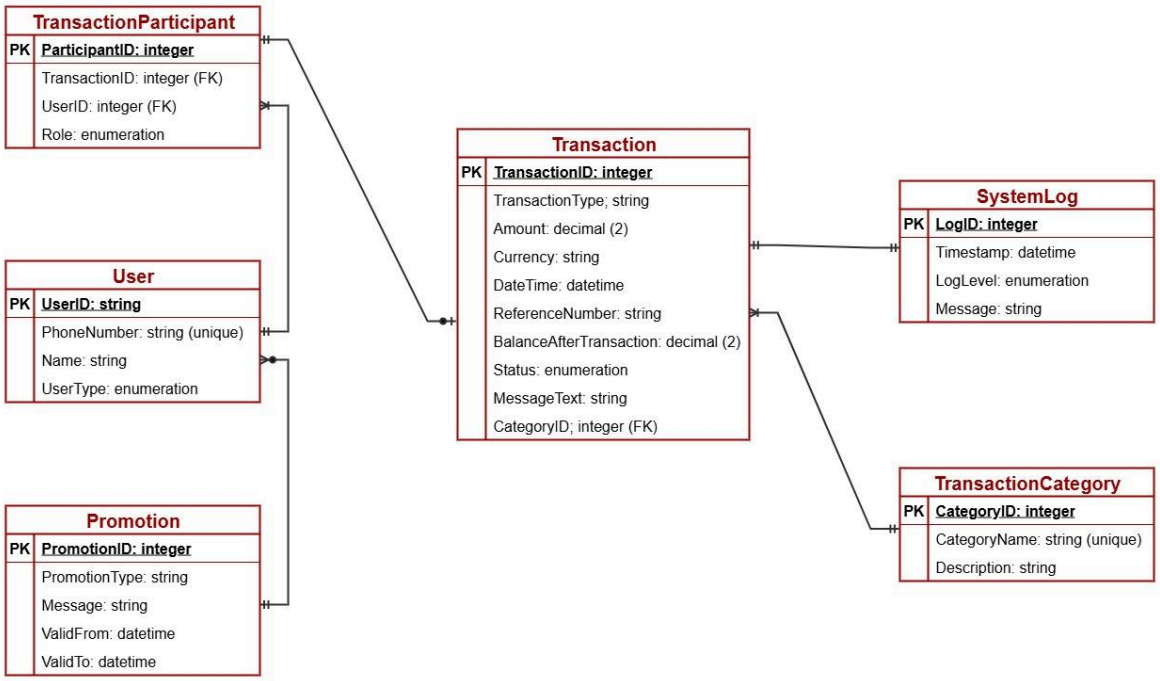
MoMo (Mobile Money) Enterprise ERD



This ERD illustrates the intricate connections between key components necessary for a mobile money system's functionality:

- The *Transaction* entity retains crucial attributes such as TransactionType, Amount, currency, and status for comprehensive tracking of financial operations.
- The *TransactionParticipant* entity links each transaction to its users while specifying each participant's role in the transaction, enhancing accountability and enabling detailed transaction review.
- The *User* entity differentiates and manages users effectively based on identification details and user type (sender, receiver, agent).
- The *SystemLog* entity maintains system integrity, logging transactions for in-depth auditing, real-time monitoring, and troubleshooting.
- The *TransactionCategory* entity facilitates categorization, enabling transactions to be organized into coherent groups for efficient reporting and analysis.
- The *Promotion* entity handles promotions (e.g., airtime top-ups), fostering user engagement through targeted offers based on user-specific criteria.

The relationships between these entities form a robust structure that supports scalability, ensuring the enterprise can expand without sacrificing control or efficiency. These design decisions ensure the system remains reliable and flexible, meeting both immediate and future needs of mobile money operations by providing a solid foundation for transaction management, user interaction, and system integrity.

Key Relations

From Entity	**Relationship Type**	**To Entity**	**Description**
Transaction	1-to-many	TransactionParticipant	A transaction can involve multiple participants (sender, receiver, agent)
User	1-to-many	TransactionParticipant	A user may participate in many transactions
Transaction	many-to-1	TransactionCategory	Each transaction belongs to one category, like 'deposit' or 'payment'
SystemLog	independent	-	Logs system events separately
Promotion	independent	-	Manages promotions and alerts

Data Dictionary

The mobile money database design has the entities: User, TransactionCategory, Transaction, TransactionParticipant, SystemLog, and Promotion. Below are the contents, format, and structure of the database and the relationship between its elements, used to control access to it and its manipulation.

User Table

Stores mobile money account owners and contacts involved in transactions.

Column	Data Type	Data Size	Description	Example
UserID	INT	Flexible	Unique identifier for each user	3
PhoneNumber	VARCHAR	20	User phone number, unique identifier	+250782345678
Name	VARCHAR	255	User full name	Alice Johnson
UserType	ENUM	Flexible	Role of user (sender/receiver/ agent) in transaction	agent

TransactionCategory Table

Categorizes transactions for reporting and analytics.

Column	Data Type	Data Size	Description	Example
CategoryID	INT	Flexible	Unique identifier for transaction category	4
CategoryName	VARCHAR	50	Category name, e.g, payment, transfer, deposit	withdrawal
Description	TEXT	Flexible	Description of the category of the transaction	Cash withdrawn from account

Transaction Table

Mobile money transaction records.

Column	Data Type	Data Size	Description	Example
TransactionID	INT	Flexible	Unique transaction record identifier	10
TransactionType	VARCHAR	50	Type of transaction, e.g., deposit, payment	withdrawal
Amount	DECIMAL	(15, 2)	Transaction amount	30000.00
Currency	VARCHAR	10	Currency code, e.g, USD	RWF
DateTime	DATETIME	Flexible	Timestamp of the transaction	2025-09-13 11:00:00
ReferenceNumber	VARCHAR	100	Payment token or voucher number	WDL11223
BalanceAfterTransaction	DECIMAL	(15, 2)	Account balance after transaction	50000.00
Status	ENUM	Flexible	Status of transaction (confirmed/failed/pending)	pending
MessageText	TEXT	Flexible	Full original SMS message content	Withdrawal request of 30000 RWF
CategoryID	INT	Flexible	Foreign key to transaction category	4

TransactionParticipant Table

Participants involved in each transaction.

Column	Data Type	Data Size	Description	Example
ParticipantID	INT	Flexible	Unique participant entry identifier	5
TransactionID	INT	Flexible	Foreign key to transaction	3
UserID	INT	Flexible	Foreign key to user	4
Role	ENUM	Flexible	Participant role in transaction (sender/receiver/agent)	sender

SystemLog Table

Logs related to system processes like ETL, validation, and errors.

Column	Data Type	Data Size	Description	Example
LogID	INT	Flexible	Unique system log identifier	4
Timestamp	DATETIME	Flexible	Log entry timestamp	2025-09-15 09:03:00
LogLevel	ENUM	Flexible	Level of log severity (info/warning/error)	error
Message	TEXT	Flexible	Log message content	Failed to process transaction 6

Promotion Table

Promotions or security alerts associated with transactions.

Column	Data Type	Data Size	Description	Example
PromotionID	INT	Flexible	Unique promotion identifier	4
PromotionType	VARCHAR	50	Type of promotion or alert	promotion
Message	TEXT	Flexible	Promotion or alert message content	Refer a friend and earn 5000 RWF
ValidFrom	DATE	Flexible	Promotion start date	2025-09-10
ValidTo	DATE	Flexible	Promotion end date	2025-10-10

Unique Rules and Database Implementation

The mobile money database design has unique rules to enhance security and accuracy. Its implementation is done using sample queries (CRUD operations), demonstrating its functionality.

The unique rule used in the database is the UNIQUE constraint in:

- User table:
    - The PhoneNumber column has a UNIQUE constraint to ensure that each phone number is distinct, serving as a unique identifier across transactions.
- In the TransactionCategory table:
    - The CategoryName column has a UNIQUE constraint ensuring that each transaction category name is unique. This avoids duplicate category names like "deposit," "payment," etc.

```
10  -- Users Table
11  CREATE TABLE User (
12      UserID INT PRIMARY KEY AUTO_INCREMENT COMMENT 'Auto-generated ID for each user to internally track and link records',
13      PhoneNumber VARCHAR(20) NOT NULL UNIQUE COMMENT 'This is User's phone number to serves as their unique identifier across transactions',
14      Name VARCHAR(255) COMMENT 'User full name',
15      UserType ENUM('sender', 'receiver', 'agent') NOT NULL COMMENT 'Role of user in transaction'
16  ) COMMENT='User table to store mobile money account owners and contacts involved in transactions';
17
```

```
28
29  -- Categories(Transaction) Table
30  CREATE TABLE TransactionCategory (
31      CategoryID INT PRIMARY KEY AUTO_INCREMENT COMMENT 'Unique identifier for transaction category',
32      CategoryName VARCHAR(50) NOT NULL UNIQUE COMMENT 'Category name to store deposit, payment, transfer, and ...',
33      Description TEXT COMMENT 'Description about the category'
34  ) COMMENT='TransactionCategory table for categorizing transactions for reporting and analytics';
35
```

The sample queries (Create, Read, Update, and Delete (CRUD) operations) used to demonstrate the functionality of the database are:

- SHOW TABLES; to list tables in the database
- CREATE TABLE IF NOT EXISTS *table_name* (*column1 datatype*, *column2 datatype*, *column3 datatype*,…); to create an entity and its attributes
- DROP TABLE *table_name*; to delete an entity
- INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...) VALUES (*value1*, *value2*, *value3*, ...); to insert data into a table
- UPDATE *table_name* SET *column1* = *value1*, *column2* = *value2*, … WHERE *condition*; to modify the existing records in a table

- SELECT *column1*, *column2, ...* FROM *table_name* WHERE *condition*; ORDER BY *column1, column2, ...* DESC; to select data from the database in descending order

Below are visuals of the queries in the command line and the graphical user interfaces (CLI & GUI):

CLI VIEW	GUI VIEW BEFORE CRUD	GUI VIEW AFTER CRUD