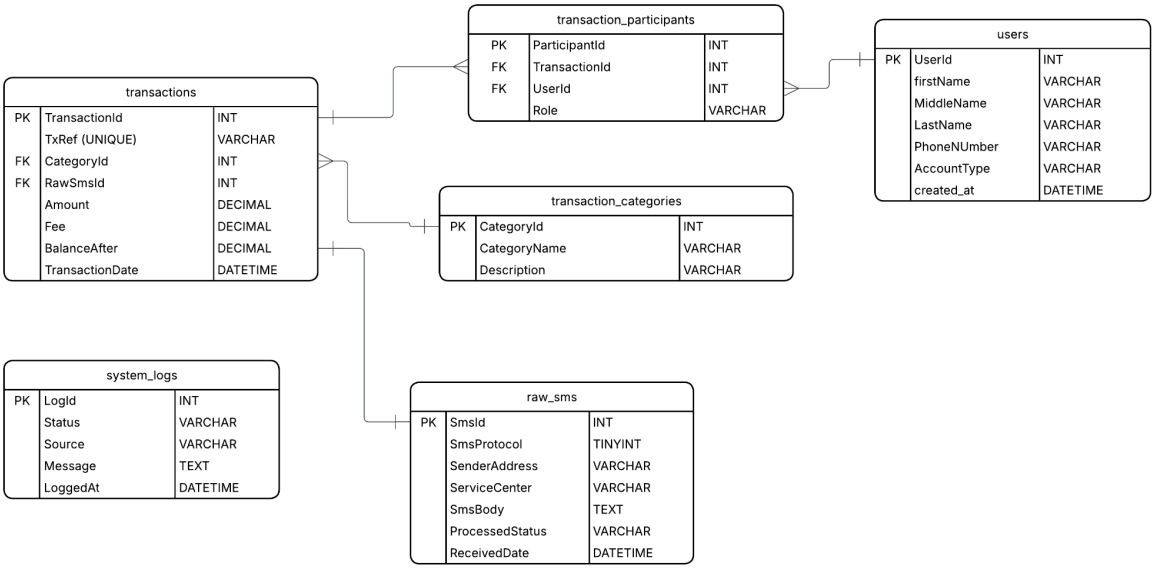
Entity Relationship Diagram (Team 11)

Lucid Chart Link



Design Process Explanation

When we went to design this database, we wanted to ensure that every piece of a MoMo SMS could be saved properly without losing any critical data. We started by looking at the XML data and realized that a single text message actually has a lot of different parts, like the transaction ID, the amount, and the people involved.

So we decided to create a table called raw_sms to hold the original text exactly how it comes in(most of it anyway). We did this because if something goes wrong when we are trying to process the data, we can go back to read the original message.

The tricky part was figuring out how to handle transaction tables and users but since one transaction always involves two people (like Jane Smith and Robert Brown in XML), we used a "junction table" called transaction_participants. This allowed us to link one transaction to two different users and give them "roles" like sender or receiver.

To properly normalize our dbs we decided to implement foreign keys wherever was appropriate so we also made a transaction_categories table. This helps the system know if a transaction is a simple transfer, a bank deposit, or a payment. We also added a system_logs table because, in a real system, you need a way to track errors or see if the data processing is actually working. And finally for the money, like the balance and fees, we used the DECIMAL type to make sure the math stays exact and doesn't have rounding errors.

One last conflict was knowing whether to store the user’s balance we create an account table with their user_id, account number and balance but we ended up adding that column to the transactions table instead and calling it “balance_after”.

Tables identified from XML data structure document shared:

Tables / Entities	Why they matter
raw_sms	It stores the original xml sms data
transactions	Financial events extracted from SMS
transaction_categories	For transaction type categories
users	Stakeholders in the system
transaction_participants (junction table)	This table will connect users and their transactions
system_logs	It's for tracking actions called out in the system

Tables Entities Defined:

1. **Raw_sms**

Column	DataType	Constraints	Description
SmsId	INT	Primary Key, Auto Increment	Unique id for every received sms
SmsProtocol	TINYINT	UNIQUE, NOT NULL	Protocol value extracted from the xml
SenderAddress	VARCHAR	FOREIGN KEY	Sender of the sms
ServiceCenter	VARCHAR	FOREIGN KEY	SMS service center number
SmsBody	TEXT	CHECK(amount >= 0)	Message body content
ProcessedStatus	VARCHAR	Default 0	Flag to track if the message was parsed yet.
ReceivedDate	DATETIME	NULL Allowed	Human readable date and time for when the user received sms
created_at	DATETIME	NOT NULL	Date and time this entry was made into the db

2. Users

Column	DataType	Constraints	Description
UserId	INT	Primary Key, Auto Increment	This is the primary key for this table to uniquely identify each user
firstName	VARCHAR	NOT NULL	First name extracted from sms
MiddleName	VARCHAR	NULL ALLOWED	Second name extracted from sms
LastName	VARCHAR	NOT NULL	Last name extracted from sms
PhoneNumber	VARCHAR	NULL ALLOWED	Stores phone numbers if they are available
AccountType	VARCHAR	‘Customer’, ‘merchant’ or ‘system’	Stores the role of the user
created_at	DATETIME	NOT NULL	Date and time of the record entry in the db

3. Transaction_categories

Column	DataType	Constraints	Description
CategoryId	INT	Primary Key, Auto Increment	This is the primary key for this table to uniquely identify categories
CategoryName	VARCHAR	UNIQUE, NOT NULL	This carries the category name
Description	VARCHAR	FOREIGN KEY	Explanation of the transaction type

4. Transactions

Column	DataType	Constraints	Description

TransactionId	INT	Primary Key, Auto Increment	This is the primary key for this table to uniquely identify the transaction records.
TxRef	VARCHAR	UNIQUE, NOT NULL	Transaction reference number (TxId)
CategoryId	INT	FOREIGN KEY	Links to transaction category
RawSmsId	INT	FOREIGN KEY	Links transactions to the original sms source
Amount	DECIMAL(12,2)	CHECK(amount >= 0)	Amount transferred or paid
Fee	DECIMAL	Default 0	Transaction fee charged
BalanceAfter	DECIMAL	NULL Allowed	Balance after transaction
TransactionDate	DATETIME	NOT NULL	Date and time of the transaction
created_at	DATETIME	NOT NULL	Date and time this entry was made into the db

5. Transactions_Participants

Column	**DataType**	**Constraints**	**Description**
ParticipantId	INT	Primary Key, Auto Increment	This column is for unique identification of the primary key for this table.
TransactionId	INT	PRIMARY KEY, FOREIGN KEY	Transaction reference number (TxId)
UserId	INT	PRIMARY KEY, FOREIGN KEY	References the user involved
Role	VARCHAR	‘sender’,’receiver’	It is an enum types and it defines if the user sent or received the money

6. System_logs

Column	DataType	Constraints	Description
LogId	INT	Primary Key, Auto Increment	This is the primary key for this table to uniquely identify the transaction records.
Status	VARCHAR	‘info’,’warn’,’error’	Enum type, helps developers access system failure level
Source	VARCHAR	Null Allowed	Will be useful for error handling to know where the error came from whether backend code or from parsing the sms.
Message	TEXT	Not null	Description of what happened
LoggedAt	DATETIME	DEFAULT CURRENT_TIME STAMP	Automatic timestamp of the log entry

Sample queries demonstrating our database functionality

1.

```
mysql> INSERT INTO users (first_name, last_name, phone_number, account_type)
    -> VALUES ('Eric', 'Nsengimana', '+250780111222', 'customer');
Query OK, 1 row affected (0.18 sec)

mysql> SELECT * FROM users WHERE phone_number = '+250780111222';
+---------+------------+-------------+------------+--------------+--------------+---------------------+
| user_id | first_name | middle_name | last_name  | phone_number | account_type | created_at          |
+---------+------------+-------------+------------+--------------+--------------+---------------------+
|       4 | Eric       | NULL        | Nsengimana | +250780111222 | customer     | 2026-02-11 11:43:28 |
+---------+------------+-------------+------------+--------------+--------------+---------------------+
1 row in set (0.04 sec)

mysql>
```

2.

```
mysql> SELECT * FROM users WHERE phone_number = '+250780111222';
+---------+------------+-------------+------------+--------------+--------------+---------------------+
| user_id | first_name | middle_name | last_name  | phone_number | account_type | created_at          |
+---------+------------+-------------+------------+--------------+--------------+---------------------+
|       4 | Eric       | NULL        | Nsengimana | +250780111222 | customer     | 2026-02-11 11:43:28 |
+---------+------------+-------------+------------+--------------+--------------+---------------------+
1 row in set (0.14 sec)
```

3.

```
mysql> INSERT INTO users (first_name, last_name, phone_number, account_type)
    -> VALUES ('Test', 'Failure', '+250780000000', 'superuser');
ERROR 3819 (HY000): Check constraint 'check_account_type' is violated.
mysql>
```

4.

```
mysql> SELECT
    -> t.tx_reference,
    -> u.first_name AS sender,
    -> t.amount,
    -> tc.category_name
    -> FROM transactions t
    -> JOIN transaction_participants tp ON t.transaction_id = tp.transaction_id
    -> JOIN users u ON tp.user_id = u.user_id
    -> JOIN transaction_categories tc ON t.category_id = tc.category_id
    -> WHERE tp.role = 'sender';
+--------------+--------+----------+---------------+
| tx_reference | sender | amount   | category_name |
+--------------+--------+----------+---------------+
| TXN001       | John   | 50000.00 | Transfer      |
+--------------+--------+----------+---------------+
1 row in set (0.15 sec)

mysql>
```

5.

```
mysql> UPDATE users
    -> SET phone_number = '+250780999888'
    -> WHERE last_name = 'Nsengimana';
Query OK, 1 row affected (0.11 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM users WHERE last_name = 'Nsengimana';
+---------+------------+-------------+------------+---------------+--------------+---------------------+
| user_id | first_name | middle_name | last_name  | phone_number  | account_type | created_at          |
+---------+------------+-------------+------------+---------------+--------------+---------------------+
|       4 | Eric       | NULL        | Nsengimana | +250780999888 | customer     | 2026-02-11 11:43:28 |
+---------+------------+-------------+------------+---------------+--------------+---------------------+
1 row in set (0.01 sec)

mysql>
```

6.

```
mysql> INSERT INTO transactions (tx_reference, amount, transaction_date)
    -> VALUES ('ERROR_TXN', -500.00, NOW());
ERROR 3819 (HY000): Check constraint 'check_positive_amount' is violated.
mysql>
```

7.

```
mysql> SELECT * FROM system_logs WHERE status = 'error';
Empty set (0.03 sec)

mysql>
```

8.

```
mysql> SELECT
    -> category_id,
    -> SUM(amount) as total_volume,
    -> COUNT(*) as num_tx
    -> FROM transactions
    -> GROUP BY category_id;
+-------------+--------------+--------+
| category_id | total_volume | num_tx |
+-------------+--------------+--------+
|           1 |     50000.00 |      1 |
+-------------+--------------+--------+
1 row in set (0.03 sec)

mysql>
```

9.

```
mysql> DELETE FROM system_logs WHERE message LIKE '%test%';
Query OK, 0 rows affected (0.02 sec)

mysql>
```

10.

```
mysql> SELECT 'Users' as TableName, COUNT(*) as RecordCount FROM users
    -> UNION
    -> SELECT 'Transactions', COUNT(*) FROM transactions
    -> UNION
    -> SELECT 'Logs', COUNT(*) FROM system_logs;
+-------------+-------------+
| TableName   | RecordCount |
+-------------+-------------+
| Users       |           4 |
| Transactions |           1 |
| Logs        |           0 |
+-------------+-------------+
3 rows in set (0.05 sec)

mysql>
```