Team: Yosuke Tobe, Michael Peluso, Hirthick Murugan

Goal of this phase of the project:

- Run SQL commands for creating tables including primary keys, secondary keys and foreign keys. Run SQL command files that populate each table.
- To develop an application system for the WALLET payment network database.
- Revisions from Phase 2 Deliverable: Using the relational schema given on canvas.

Description of implementation: using Flask, PHPMyAdmin, Python, and HTML:

1. Database Schema Creation

The database schema for this application has the following tables:

• WALLET_ACCOUNT:

- o Fields: SSN, Name, PhoneNo, Balance, BankID, BANumber, BAVerified
- Stores user information such as Social Security Number, contact details, and wallet balance.

• EMAIL ADDRESS:

- o Fields: EmailAdd, SSN, Verified
- Links email addresses to users.

• ELEC ADDRESS:

- o Fields: Identifier, Verified, Type
- Stores email and phone identifiers for electronic communication.

• BANK ACCOUNT:

- Fields: BankID, BANumber, Verified
- Stores linked bank account details.

• SEND TRANSACTION:

- Fields: Tld, Identifier, L_DTime, C_DTime, Memo, CReason, CType, Amount, SSN
- Tracks funds sent by users.

REQUEST_TRANSACTION:

- Fields: Rld, Amount, DateTime, Memo, SSN
- o Tracks fund requests initiated by users.

• REQUEST_FROM:

- Fields: Rld, Identifier, Percentage
- o Links a money request transaction to the specific user who initiated the request.

2. Database Instance Creation

Using **PHPMyAdmin**, the schema can be created by executing SQL CREATE TABLE statements based on the schema description above. Each table includes relevant constraints such as primary keys and foreign keys.

3. Application Programs

Backend: Flask Application

- File Example: app.py
- Purpose: Handles user interactions, database operations, and routing.
- Key Features:
 - User Signup: Validates user input (SSN, email, phone), ensures no duplication, and stores data in the WALLET_ACCOUNT and EMAIL_ADDRESS tables.
 - Login: Authenticates users based on SSN.
 - Dashboard: Displays user details, balance, and transaction history using Flask templates.
 - Fund Management: Allows adding or deducting funds.
 - Transactions: Supports sending and requesting money, with validation and database updates.
 - o Bank Linking: Facilitates linking and unlinking bank accounts.

Frontend: HTML Templates

- File Example: dashboard.html
- Purpose: Provides a user-friendly dashboard interface.
- Key Features:
 - Displays user information such as name, SSN, phone number, and wallet balance.
 - Lists recent transactions (both sent and requested).
 - Styled with Bootstrap for responsiveness.

Database Interaction:

- Flask interacts with the MySQL database using helper functions (execute_query and fetch query) for executing and retrieving SQL queries.
- Error handling ensures application stability during database interactions.

Problems faced:

- What technologies to use:
 - Solved by going with what was comfortable for most teammates.
- Contribution of each member:
 - Resolved by identifying and discussing our strengths.
- 2. The SQL commands that create your tables

```
CREATE TABLE BANK_ACCOUNT (
BankID INT NOT NULL,
BANumber VARCHAR(20) NOT NULL,
Verified BOOLEAN DEFAULT FALSE,
```

```
PRIMARY KEY (BankID, BANumber) -- Composite Primary Key
);
-- Create the ELEC ADDRESS table
CREATE TABLE ELEC ADDRESS (
    Identifier VARCHAR(100) PRIMARY KEY, -- Central electronic address
(email/phone)
    Verified BOOLEAN DEFAULT FALSE,
    Type ENUM('Email', 'Phone') NOT NULL
);
-- Create the WALLET ACCOUNT table
CREATE TABLE WALLET ACCOUNT (
    SSN VARCHAR(11) PRIMARY KEY, -- Unique identifier for wallet account
    Name VARCHAR (50) NOT NULL,
    PhoneNo VARCHAR (15) UNIQUE NOT NULL,
    Balance DECIMAL(10, 2) DEFAULT 0,
    BankID INT, -- Foreign key part of composite key
    BANumber VARCHAR(20), -- Foreign key part of composite key
    BAVerified BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (BankID, BANumber) REFERENCES BANK ACCOUNT (BankID,
BANumber),
    FOREIGN KEY (PhoneNo) REFERENCES ELEC ADDRESS (Identifier)
);
-- Create the EMAIL ADDRESS table
CREATE TABLE EMAIL ADDRESS (
    EmailAdd VARCHAR(100) PRIMARY KEY, -- Email is the primary key
    SSN VARCHAR(11) NOT NULL, -- Foreign key to WALLET ACCOUNT.SSN
    Verified BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (SSN) REFERENCES WALLET ACCOUNT (SSN),
    FOREIGN KEY (EmailAdd) REFERENCES ELEC ADDRESS (Identifier)
);
-- Create the SEND TRANSACTION table
CREATE TABLE SEND TRANSACTION (
    TId INT AUTO INCREMENT PRIMARY KEY, -- Unique transaction ID
    Identifier VARCHAR(100) NOT NULL, -- Recipient's electronic address
    L DTime DATETIME NOT NULL, -- Initiated timestamp
    C DTime DATETIME, -- Completed timestamp
    Memo TEXT,
    CReason TEXT, -- Cancellation reason
    CType ENUM('Pending', 'Completed', 'Cancelled') NOT NULL, --
Transaction status
    Amount DECIMAL(10, 2) NOT NULL,
    SSN VARCHAR(11) NOT NULL, -- Sender's SSN
```

```
FOREIGN KEY (Identifier) REFERENCES ELEC ADDRESS(Identifier), --
Recipient reference
    FOREIGN KEY (SSN) REFERENCES WALLET ACCOUNT(SSN) -- Sender reference
);
-- Create the REQUEST TRANSACTION table
CREATE TABLE REQUEST TRANSACTION (
    RId INT AUTO INCREMENT PRIMARY KEY, -- Unique request ID
    Amount DECIMAL(10, 2) NOT NULL,
    DateTime DATETIME NOT NULL,
   Memo TEXT,
    SSN VARCHAR(11) NOT NULL, -- Requester's SSN
    FOREIGN KEY (SSN) REFERENCES WALLET ACCOUNT(SSN)
);
-- Create the REQUEST FROM table
CREATE TABLE REQUEST FROM (
    RId INT NOT NULL, -- Request ID
    Identifier VARCHAR(100) NOT NULL, -- Recipient's electronic address
    Percentage DECIMAL(5, 2), -- Percentage of amount
    PRIMARY KEY (RId, Identifier),
    FOREIGN KEY (RId) REFERENCES REQUEST TRANSACTION(RId), -- Request
reference
    FOREIGN KEY (Identifier) REFERENCES ELEC ADDRESS(Identifier) --
Recipient reference
);
3. The SQL commands that populate your tables
INSERT INTO BANK ACCOUNT (BankID, BANumber, Verified)
VALUES
(1, '1234567890', TRUE),
(2, '9876543210', FALSE);
-- Insert sample data into ELEC ADDRESS
INSERT INTO ELEC ADDRESS (Identifier, Verified, Type)
VALUES
('leonardo.dicaprio@example.com', TRUE, 'Email'),
('meryl.streep@example.com', TRUE, 'Email'),
('5551234567', TRUE, 'Phone'),
('5559876543', FALSE, 'Phone');
-- Insert sample data into WALLET ACCOUNT
```

```
INSERT INTO WALLET ACCOUNT (SSN, Name, PhoneNo, Balance, BankID, BANumber,
BAVerified)
VALUES
('123456789', 'Leonardo DiCaprio', '5551234567', 1000.00, 1, '1234567890',
('987654321', 'Meryl Streep', '5559876543', 500.00, 2, '9876543210',
FALSE);
-- Insert sample data into EMAIL ADDRESS
INSERT INTO EMAIL ADDRESS (EmailAdd, SSN, Verified)
VALUES
('leonardo.dicaprio@example.com', '123456789', TRUE),
('meryl.streep@example.com', '987654321', TRUE);
-- Insert sample data into SEND TRANSACTION
INSERT INTO SEND TRANSACTION (Identifier, L DTime, C DTime, Memo, CReason,
CType, Amount, SSN)
VALUES
('meryl.streep@example.com', NOW(), NOW(), 'Payment for groceries', NULL,
'Completed', 100.00, '123456789'),
('5559876543', NOW(), NULL, 'Lunch split', NULL, 'Pending', 50.00,
'123456789');
-- Insert sample data into REQUEST TRANSACTION
INSERT INTO REQUEST TRANSACTION (Amount, DateTime, Memo, SSN)
VALUES
(200.00, NOW(), 'Utilities', '123456789'),
(150.00, NOW(), 'Dinner', '987654321');
-- Insert sample data into REQUEST FROM
INSERT INTO REQUEST FROM (RId, Identifier, Percentage)
VALUES
(1, 'leonardo.dicaprio@example.com', 50.00),
(1, 'meryl.streep@example.com', 50.00),
(2, 'leonardo.dicaprio@example.com', 60.00),
(2, '5551234567', 40.00);
```

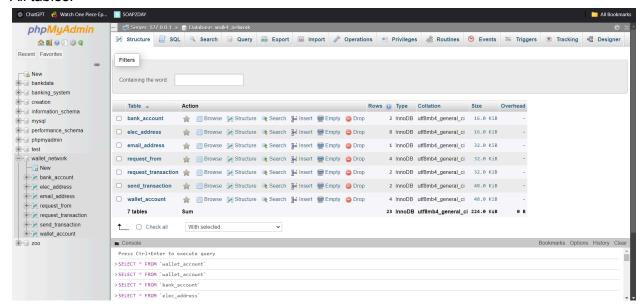
4. The source code.

https://github.com/michaelpeluso/Wallet-Netowrk

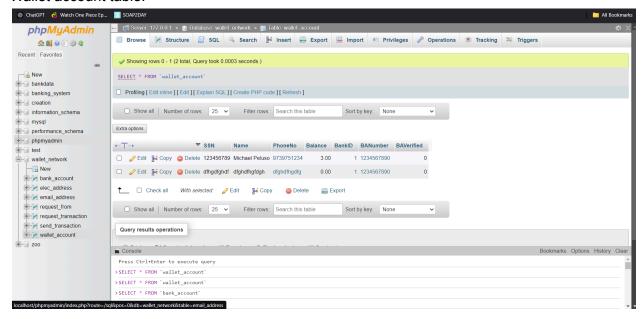
5. A printout of the use of the program (some screenshots).

Sample DB screenshots

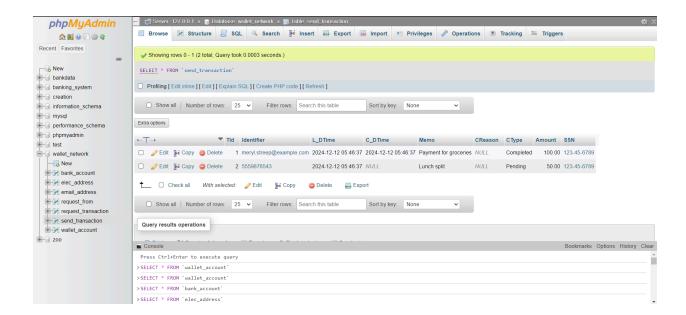
All tables:



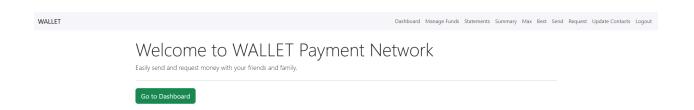
Wallet account table:

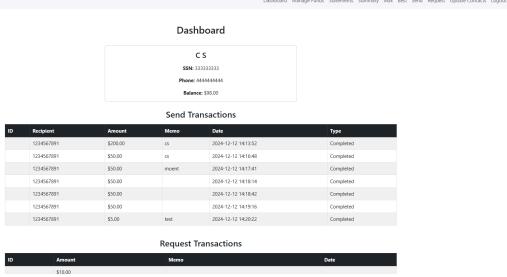


Send_transaction table:



Sample application:

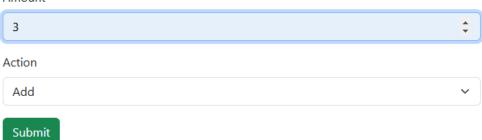




Manage Funds

Current balance: \$98.00

Amount



Send Money

Recipient email or phone		
mpdf@nasdf.edu		
Amount		
238.39		
Memo (optional)		
sending money		
Confirm		

Request Money

Recipient email or phone

1230981412

Amount

23.03

Memo (optional)

pay me please

Request

Monthly Transaction Summary

Month	Total Sent	Average Sent	Total Received	Average Received
2024-11	\$11.00	\$5.50	\$11.00	\$5.50
2024-12	\$500035.00	\$125008.75	\$500035.00	\$125008.75





Transaction Summary



Summary from 2024-12-09 to 2024-12-25

Transaction Total: \$455.00

Maximum Transactions Per Month

Month	Max Amount	Recipient
2024-12	\$200.00	1234567891

Davinoura manageranas sacemento sammary ma

Best Users

Top Sender

Total Sent: \$500046.00

Top Receiver

Total Received: \$200.00