

Team: Yosuke Tobe, Michael Peluso, Hirtheek 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:**
 - Fields: SSN, Name, PhoneNo, Balance, BankID, BANumber, BAVerified
 - Stores user information such as Social Security Number, contact details, and wallet balance.
- **EMAIL_ADDRESS:**
 - Fields: EmailAdd, SSN, Verified
 - Links email addresses to users.
- **ELEC_ADDRESS:**
 - 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: TId, Identifier, L_DTime, C_DTime, Memo, CReason, CType, Amount, SSN
 - Tracks funds sent by users.
- **REQUEST_TRANSACTION:**
 - Fields: RId, Amount, DateTime, Memo, SSN
 - Tracks fund requests initiated by users.
- **REQUEST_FROM:**
 - Fields: RId, Identifier, Percentage
 - 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.
 - **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',
TRUE),
('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:

phpMyAdmin interface showing the structure of the 'wallet_network' database. The 'Structure' tab is selected, displaying a list of tables and their properties.

Table	Action	Rows	Type	Collation	Size	Overhead
bank_account	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
elec_address	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	8	InnoDB	utf8mb4_general_ci	16.0 KiB	-
email_address	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
request_from	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
request_transaction	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
send_transaction	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	2	InnoDB	utf8mb4_general_ci	48.0 KiB	-
wallet_account	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	4	InnoDB	utf8mb4_general_ci	48.0 KiB	-
7 tables	Sum	23	InnoDB	utf8mb4_general_ci	224.0 KiB	0 B

Console output:

```
>SELECT * FROM 'wallet_account';
>SELECT * FROM 'wallet_account';
>SELECT * FROM 'bank_account';
>SELECT * FROM 'elec_address';
```

Wallet account table:

phpMyAdmin interface showing the data of the 'wallet_account' table. The 'Browse' tab is selected, displaying the table's structure and data.

	SSN	Name	PhoneNo	Balance	BankID	BANumber	BAVerified
<input type="checkbox"/>	123456789	Michael Peluso	9739751234	3.00	1	1234567890	0
<input type="checkbox"/>	dfghdghdf	dfghdghdgh	dfghdghdgh	0.00	1	1234567890	0

Console output:

```
>SELECT * FROM 'wallet_account';
>SELECT * FROM 'wallet_account';
>SELECT * FROM 'bank_account';
```

Send_transaction table:

phpMyAdmin

Server: 127.0.0.1 > Database: wallet_network > Table: send_transaction

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds)

SELECT * FROM `send_transaction`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	Tid	Identifier	L_DTime	C_DTime	Memo	CReason	CType	Amount	SSN
<input type="checkbox"/>	1	meryl.streep@example.com	2024-12-12 05:46:37	2024-12-12 05:46:37	Payment for groceries	NULL	Completed	100.00	123-45-6789
<input type="checkbox"/>	2	5559876543	2024-12-12 05:46:37	NULL	Lunch split	NULL	Pending	50.00	123-45-6789

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Console

```
Press Ctrl+Enter to execute query
>SELECT * FROM `wallet_account`
>SELECT * FROM `wallet_account`
>SELECT * FROM `bank_account`
>SELECT * FROM `elec_address`
```

Sample application:

WALLET

Dashboard Manage Funds Statements Summary Max Best Send Request Update Contacts Logout

Welcome to WALLET Payment Network

Easily send and request money with your friends and family.

Go to Dashboard

Dashboard

C S

SSN: 333333333

Phone: 4444444444

Balance: \$98.00

Send Transactions

ID	Recipient	Amount	Memo	Date	Type
	1234567891	\$200.00	cs	2024-12-12 14:13:52	Completed
	1234567891	\$50.00	cs	2024-12-12 14:16:48	Completed
	1234567891	\$50.00	moent	2024-12-12 14:17:41	Completed
	1234567891	\$50.00		2024-12-12 14:18:14	Completed
	1234567891	\$50.00		2024-12-12 14:18:42	Completed
	1234567891	\$50.00		2024-12-12 14:19:16	Completed
	1234567891	\$5.00	test	2024-12-12 14:20:22	Completed

Request Transactions

ID	Amount	Memo	Date
	\$10.00		

Manage Funds

Current balance: **\$98.00**

Amount

3

Action

Add

Submit

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

Update Contacts

Phone

4444444444Verified

Change Phone

Update Phone

Emails

cs2@gmail.comVerifiedDelete

cs@gmail.comVerifyDelete

Add New Email

Add Email

Bank Information

Bank ID

2147483647

Bank Account Number

3272234465

Bank account is not verified.

Update Bank Information

Transaction Summary

Start Date

mm/dd/yyyy

End Date

mm/dd/yyyy

Get 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

[Dashboard](#) [Manage Funds](#) [Statements](#) [Summary](#) [Help](#)

Best Users

Top Sender

Total Sent: \$500046.00

Top Receiver

Total Received: \$200.00