CS 631 PROJECT

Deliverable 3

GROUP : 11
TITLE: WALLET payment network

INTRODUCTION:

In this report, we will discuss the implementation of the WALLET payment network database application system. This is the Phase 3 deliverable , which involves creating a database schema and instance, and developing an application system to interact with the database. This report will provide a detailed description of the implementation process, problems faced, and the solutions adopted to overcome those problems.

GOAL:

The goals of this phase of the project are to establish a database schema and instance, as well as to develop an application system to communicate with the database. Users should be able to check their account information, send/receive money, view statements, search transactions, edit personal information, and do other account-related operations using the application system.

Database Schema and Instance:

The database schema was created in accordance with the standards stated in Phase 2. There are four tables in the schema: User, Account, Transaction, and Request. Primary and foreign keys define the relationships between the tables. Foreign keys connect the User table to the Account, Email, Phone, and BankAccount . Foreign keys connect the Transaction table to the User and Account tables.

The SQL commands used to create the tables, including primary keys, secondary keys, and foreign keys, are shown below:

```
CREATE TABLE user (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  fullname VARCHAR(50),
  mobileno VARCHAR(10),
  email VARCHAR(30),
  password VARCHAR(20),
```

```
  ssn VARCHAR(11) UNIQUE
);

CREATE TABLE accounts (
  account_id INT PRIMARY KEY,
  ssn VARCHAR(11) UNIQUE,
  account_type VARCHAR(20),
  balance DECIMAL(10, 2),
  FOREIGN KEY (account_ssn) REFERENCES user(ssn),
  FOREIGN KEY (account_id) REFERENCES user(id),
);



//To create account for user
INSERT INTO accounts (account_id, account_ssn, account_type, balance) VALUES
(User Id, 'User SSn', 'checking', 5000.00);



CREATE TABLE transactions (
  trans_id varchar(11) PRIMARY KEY,
  sender_id INT,
  receiver_id INT,
  amount DECIMAL(10, 2),
  time_date DATETIME,
  FOREIGN KEY (sender_id) REFERENCES accounts(account_id),
  FOREIGN KEY (receiver_id) REFERENCES accounts(account_id)
);

CREATE TABLE requests (
  request_id INT AUTO_INCREMENT PRIMARY KEY,
  from_account INT NOT NULL,
  to_account INT NOT NULL,
  amount INT NOT NULL,
  status VARCHAR(20) NOT NULL DEFAULT 'Pending',
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (from_account) REFERENCES accounts(account_id),
  FOREIGN KEY (to_account) REFERENCES accounts(account_id)
);
```

Application System:

For the WALLET payment network database, we created an application system. The following menus are available on the system:

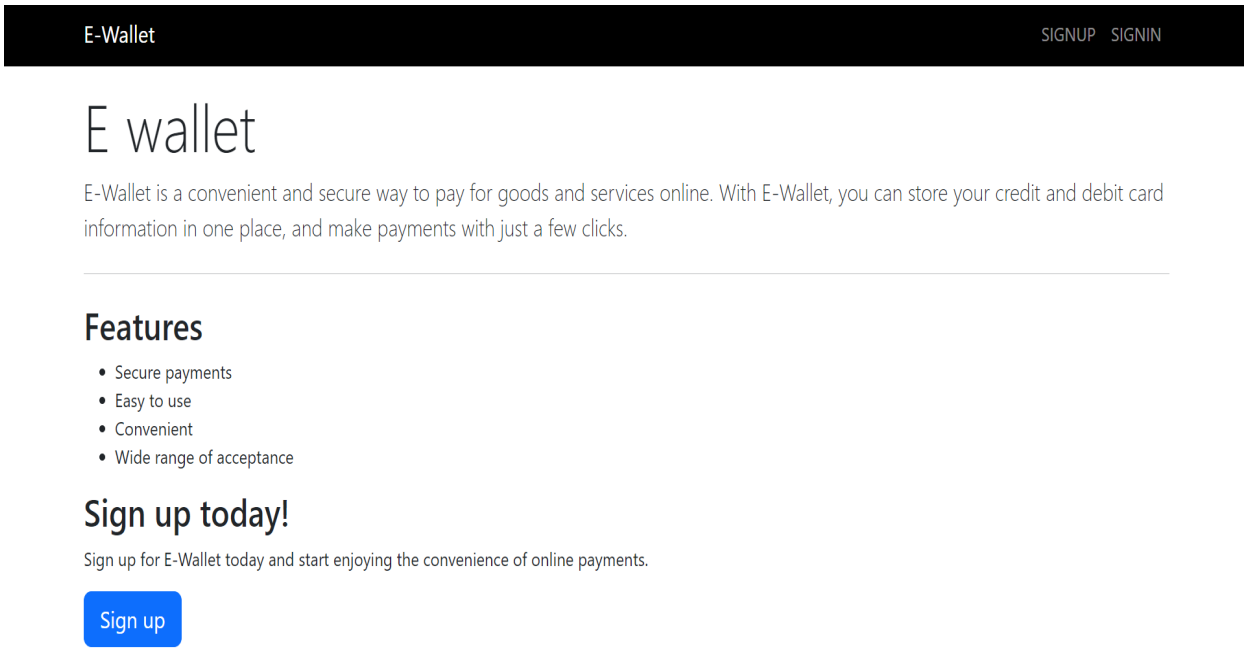
- Main Menu.
- Account Info.
- Send Money.
- Request Money.
- Statements.
- Search Transactions.
- Sign Out.

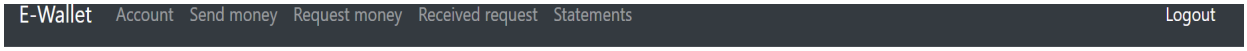
Main Menu:

The Main Menu provides users with access to all the functions of the application.

- Account info
- Send money (send money to a phone number or email address)
- Request money (request money from phone numbers and email addresses)
- Statements (total amount sent and received per month)
- Search Transactions (based on user SSN, email address, phone number, type of transaction, time_date range, etc.)
- Sign Out.

Main Menu Screenshot:

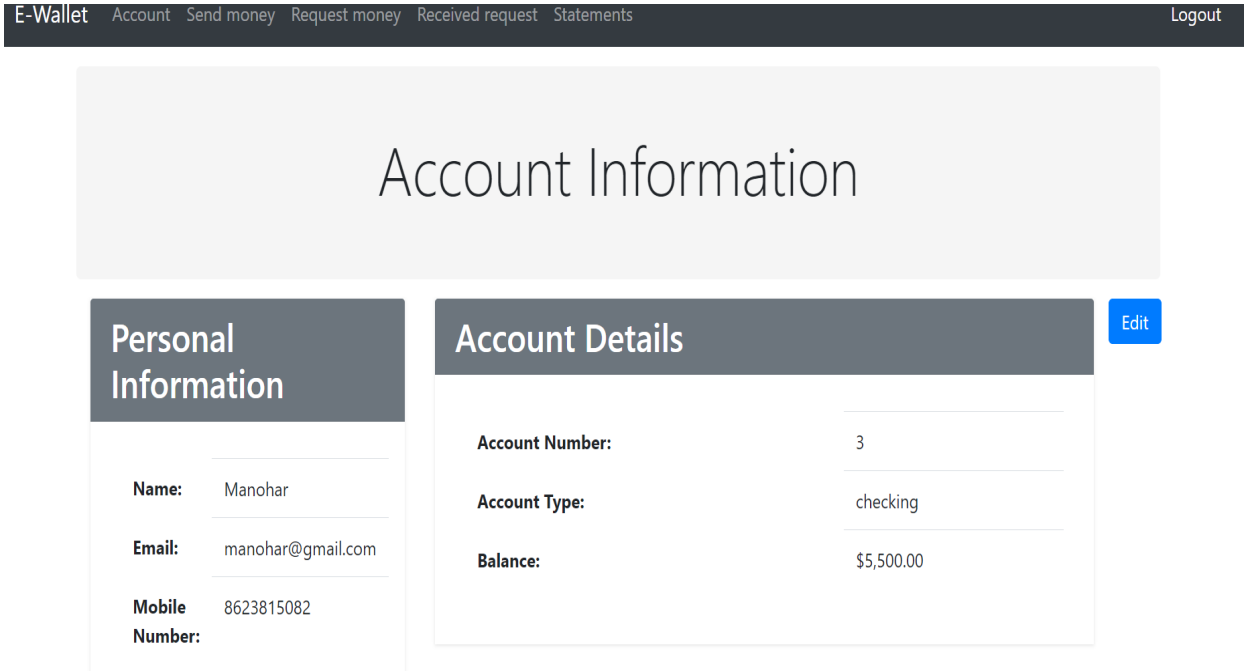


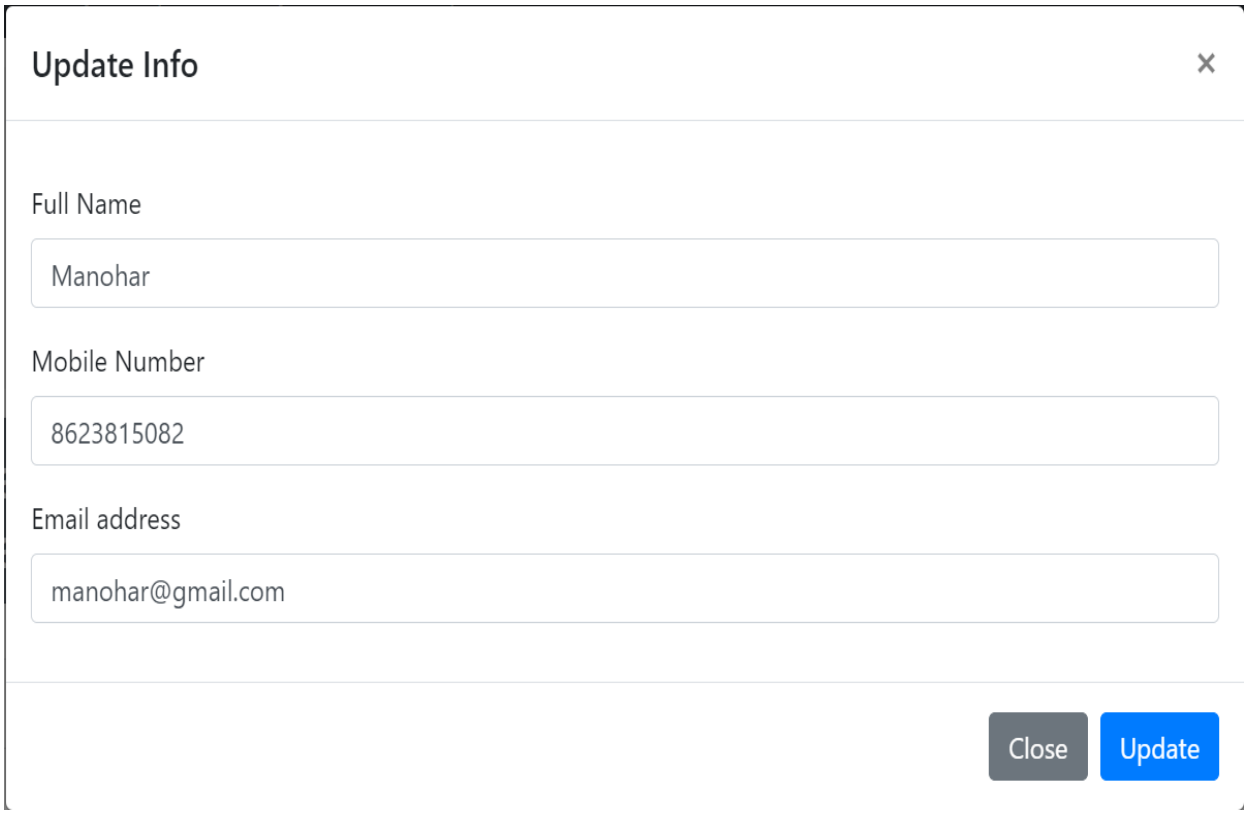


Account Info:

Account Info menu allows users to modify their personal details and add/remove contact information or bank accounts.

- Modify personal details
- Add/remove email address
- Add/remove phone number

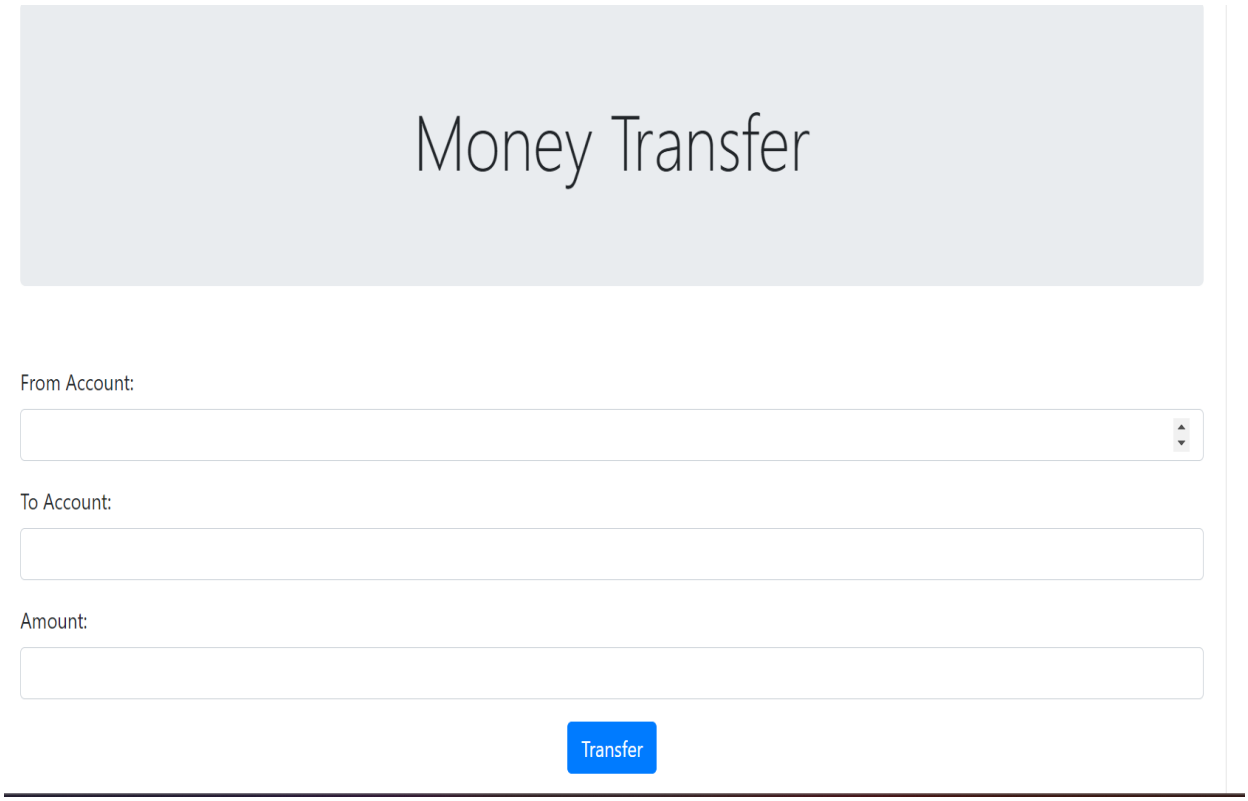

E-Wallet
Account
Send money
Request money
Received request
Statements
Logout
Account Information
Personal Information
Name:
Manohar
Email:
manohar@gmail.com
Mobile Number:
8623815082
Account Details
Edit
Account Number:
3
Account Type:
checking
Balance:
$5,500.00


Update Info
Full Name
Manohar
Mobile Number
8623815082
Email address
manohar@gmail.com
Close
Update

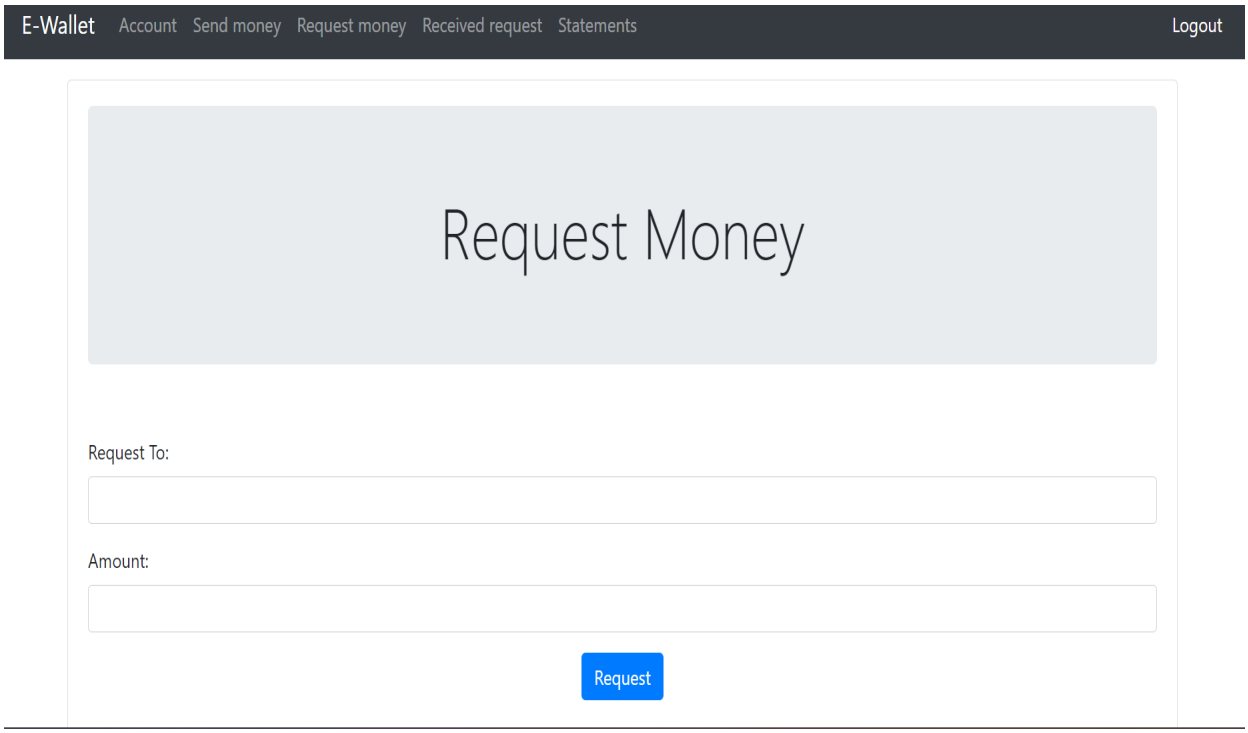
Send Money:

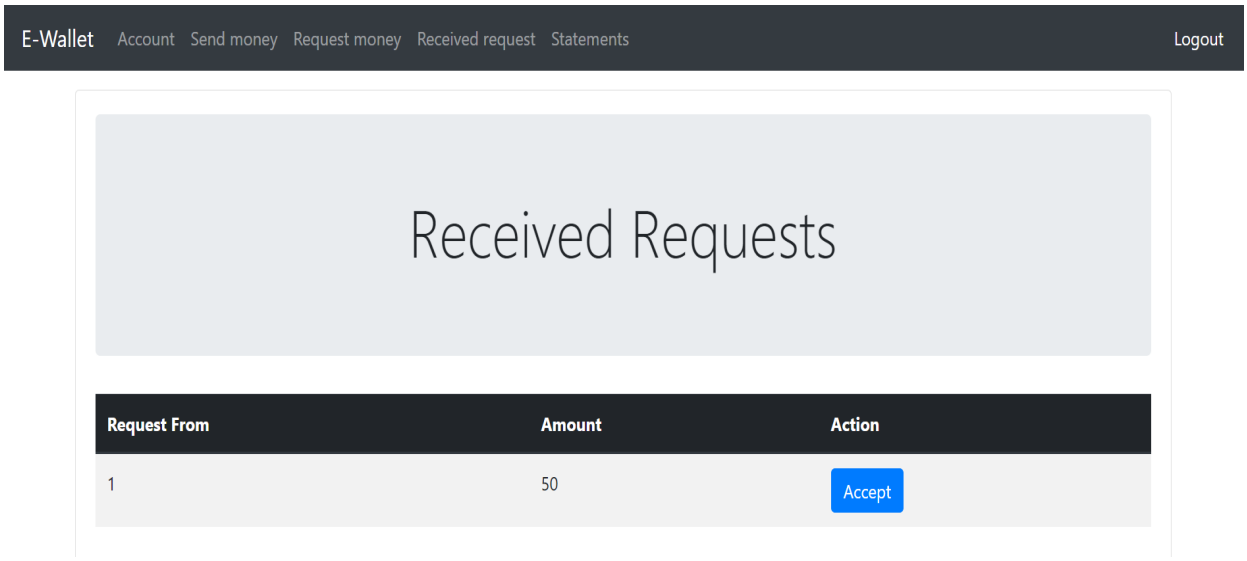
The Send Money and Request Money menus allow users to send or request money from other users by entering their From account and To account details.

Send money screen shot:



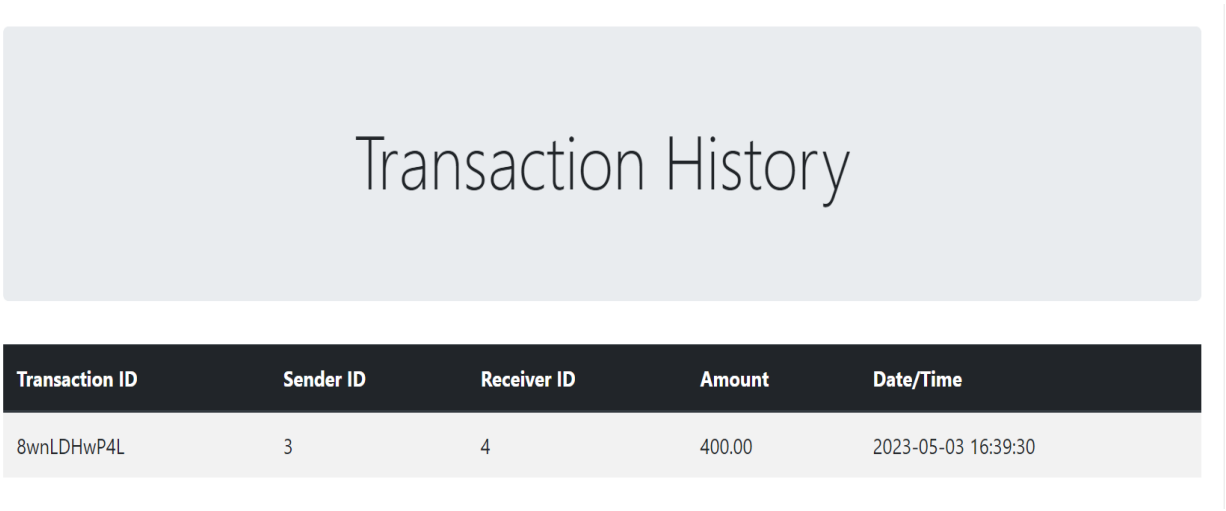
Receive money request:





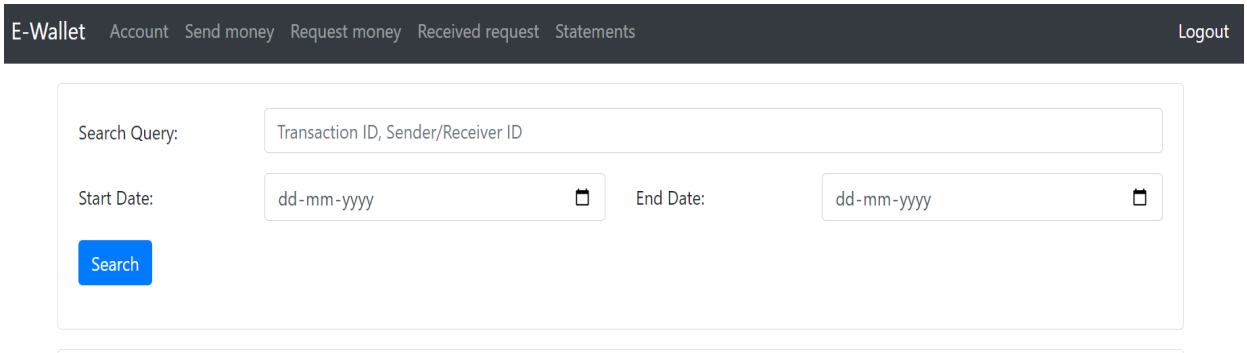
Statements:

The Statement functions provide users with a summary of their transactions, such as the total/average amount of money sent and received per month and the transactions with the maximum amount of money per month. The Best Users function lists the users who have sent or received the maximum total amount of money.



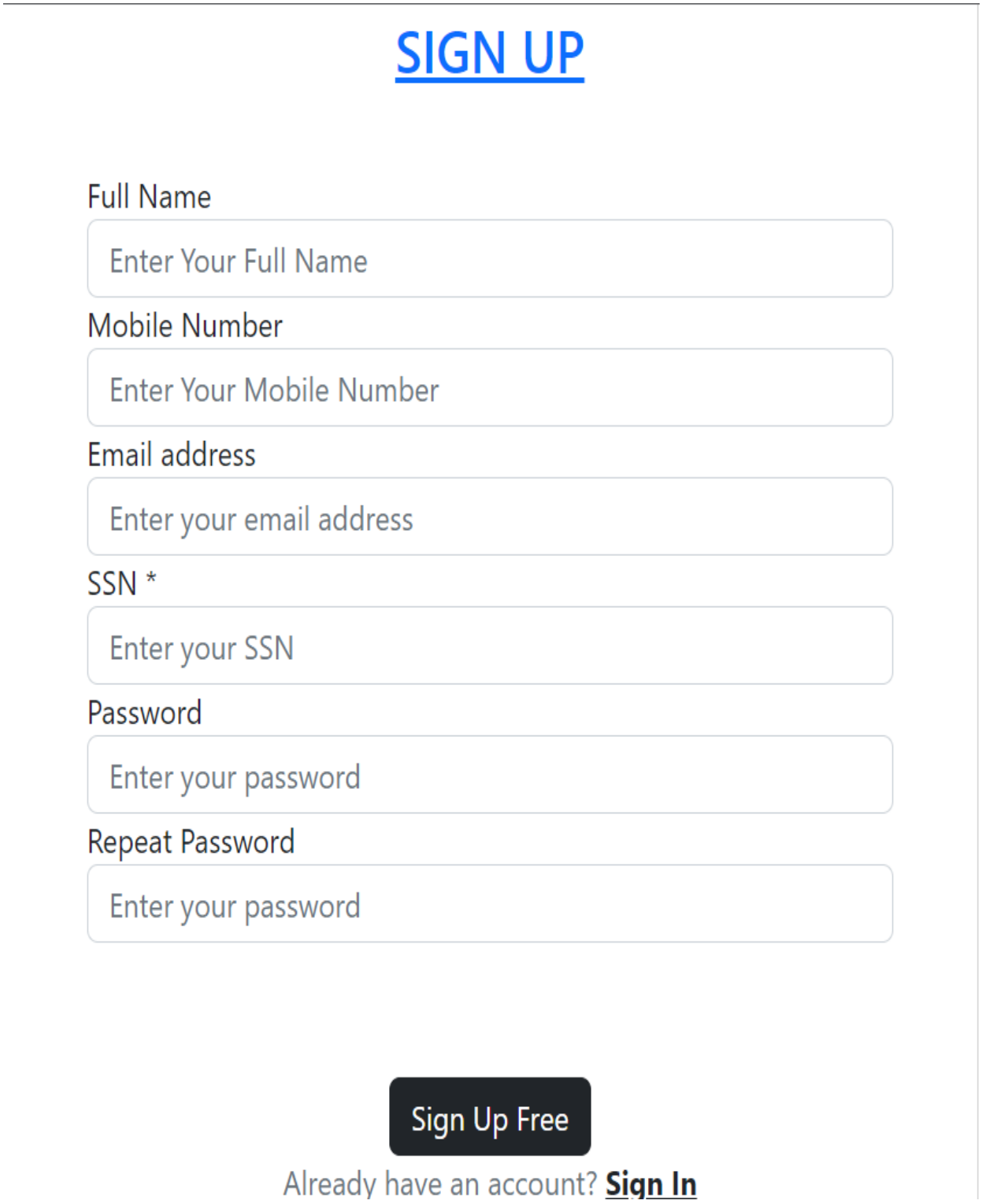
Search Transactions:

The Statements menu allows users to view their total amount sent and received per month. They can also search for transactions based on Transection Id,Sender Id/Receiver Id, and time/date range.



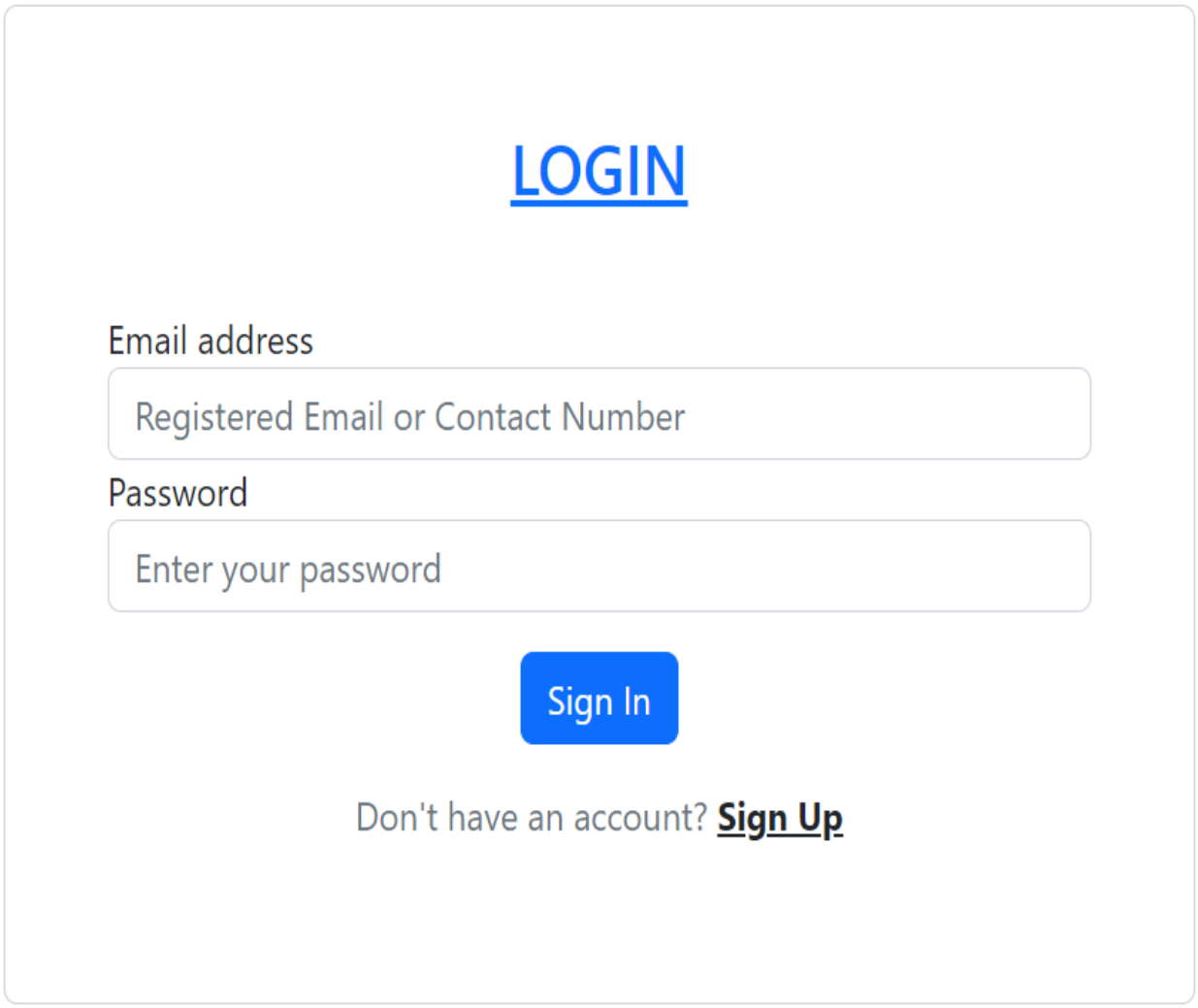
Sign up:

For creating an account, we need to register on the database by using the sign up button.



Sign In/Sign out:

After finishing the registration it shows successfully registered,then we are going to login to our account by using Email address and password.



Problems Faced and Solutions:

One of the major problems we faced during the implementation process was ensuring the correctness of the database schema and instance. We had to ensure that the relationships between the tables were correctly planned, and the SQL commands used to create the tables and populate them with data were error-free. To overcome this problem, we thoroughly tested the database schema and instance and used a variety of test cases to ensure its correctness.

Another problem we faced was designing the user interface of the application system. We had to ensure that it was easy to use and navigate for users. To overcome this problem, we conducted user testing and incorporated user feedback into the design of the application system.

Conclusion:

In conclusion, we have successfully created a database schema and instance and developed an application system for the WALLET payment network database. The application system allows users to perform a range of account-related functions, such as sending/receiving money and viewing transaction statements. We encountered several problems during the implementation process, but we were able to overcome them with careful testing and user feedback. Overall, we believe that our application system is user-friendly and meets the requirements of the project.