Database Table

# Users

* Fields
* userID
* name
* surname
* username
* email
* phoneNumber

# Keys

* Fields
* userID
* authToekn
* password

# Wallet

* Fields
* userID
* provider
* walletNumber
* type
* history (array)

# Transaction

* Fields
* senderID
* recieverID
* type
* amount
* currency
* timestamp
* status
* scamFlag

# Relationship

* User Has a Key (one user == one key)
* User has a Wallet (one user == one wallet)
* User initiates Transactions (one user == many transactions)
* Wallet has many transactions (one wallet == many transactions)

Sever Endpoints

# POST /register

* Query

JSON{

name="John";

surname="Smith";

username="johnSmith55";

phoneNumber="0795847362";

password="John12345";

email="john@example.com"

}

* Response

JSON{

userId

name

surname

username

phoneNumber

email

token

}

# POST /login

* Query

JSON{

email="john@example.com";

password="John12345"

}

* Response

JSON{

userId

username

name

surname

email

token

}

# POST /createWallet

* Query : requires “Authorization: Bearer <token>”

JSON{

provider="Capitec"

walletNumber="0679432123"

}

* Response

JSON{

Message: “Wallet successfully created

}

# POST /createTransaction

* Query : requires “Authorization: Bearer <token>”

JSON{

username: "sizwe101",

transactionType: “CapitecPay”,

amount: 500,

reference: “Utility bill”

* Response

JSON{

Message,

Status,

            }