Main Components:

1. BlockChain Module
2. UI Module
3. Integrity Module
4. Communication Module

Brief Descriptions

1. BlockChain Module

This module is the core component of our application. It maintains all blockchain operations such as creating blocks, signature and mining.

|  |
| --- |
| Block c++ |
| +blockNumber: int  +prevHash: string  +transactions: vector<transactions>  +selfHash: string  +time: tim\_t |
| +verifyHash():bool  +genHash(time: time\_t, prevHashIn: string, nonce: int):string  +addTransaction(&trans: transaction)  +addTransaction(&trans: Vector<transaction>):void  +getBlockNumber(): int  +setBlockNumber(blockNumber: int):void  +getNonce():int  +getPrevHash():string  +setPrevHash(prevHash: string):void  +getTransactions():vector<transactions>  +setTransactions((&transacts:vector<transactions>)  +getSelfHash():string  +setSelfHash(string selfHash):void  +getTime():time\_t  +setTime(time:time\_t):void  +removeLatestTransaction():void  +blockInfoToString():string  +toStringk():string  +operator<<(&stream: ostream, &in: const transaction):ostream  +transactionsToString():string |

|  |
| --- |
| blockchain c++ |
| +minedBlocks: vector<blocks> |
| +mine(blockIn: block, &key:string):string  +hexCamp(in1: string, in2: string):void  +ensureLength(in: string):string  +toString():string  +JSONOutput(witeSpaceBegginging:string, tag:string):string  +userListJSONOutput(witeSpaceBegginging:string, tag:string):string  +operator<<(&stream: ostream, &in: const transaction):ostream  +getMinedBlocks():vector<block>  +getBlockFromChain(in: int): block  +setMinedBlocks(&minedBlocks: vector<block>)  +getUserList():&vector<user>  +setUserList(&const vector<user>: listIn):void  +addToUserList(&const user: in):void  +userListToString():const string  +getByPublicKey(publicKey:string): user  +isPublicKeyInList(publicKey:string): bool  +veirfyTransactions(&in:block)const bool  +updateusers(&in:block):void |

|  |
| --- |
| Transaction c++ |
| +blockNumber: int  +senderName: string  +senderKey: string  +senderSig: string  +receiverName: string  +receiverKey: string  +amount: double  +time: time\_t  +veifySignature(): const bool |
| +getselfHash():string  +ToString():string  +operator<<(&stream: ostream, &in: const transaction):ostream  +getBlockNumber():int  +setBlockNumber(blockNumber:int):void  +setSenderName(&senderName: string):void  +setSenderKey(&senderKey: string):void  +setSendeSig(&senderSig: string):void  +getSenderName():&transaction: string  +getSenderKey():&transaction: string  +getSenderSig ():&transaction: string  +setReceiverName(&receiver: string):&transaction: string  +setReceiverKey(&receiver: string):&transaction: string  +getReceiverName():&transaction: string  +getreceiverKey():&transaction: string  +getAmount():transaction:double  +setAmount(amount:double):void  +getTime():time\_t  +setManTime(time\_t: time):void  +setAutoTime():void  +verifySignature():bool |

|  |
| --- |
| dataIntegrity c++ |
| +incrementCoin(amount:double):int  +decrementCoin(amount:double):int  +incrementUser(amount:double):int  +decrementUser(amount:double):int  +getCointAmount():double  +getUserAmount():double  +totalHeldisTotalMined():Boolean  +PHCEffect():Boolean  +userAmountNotLessThan0():boolean  +log(activity:string):void |

|  |
| --- |
| user c++ |
| +userName: string  +publicKey: string  +balance: long |
| +getUserName(): const string  +getPublicKey(): const string  +getBalance(): const long  +setUserName(in:string):void  +setPublicKey(in:string):void  +setBalance(in:long):void  +addToBalance(in:long):void  +subtractFromBalance(in:long):void  +toString(): const string  +getSetPublicKey(const &privateKeyIn:string):string  +genPrivateKey():string |

1. UI Module

This module is the point of contact for end users and it will function as the main driver for the whole application. From the UI, users will be able to create accounts, login and make transactions and logout.

1. Integrity Module

This module will ensure that no user can make transactions that they shouldn’t be able to, such as send money when their account is empty.

|  |
| --- |
| Profile c++ |
| +Create():int  +setName(name:string):int  +addAmount(amount:double):void  +subtractAmount(amount:double):void  +setPassword(inPass:string):int  +setUserName(inUserName:string):int |

1. Data Module

This module will handle all work of connecting to and operating MYSQL database. The database will keep track of users and their transactions as well as their credentials.

|  |
| --- |
| DB |
| -createTable(colNames:string, colTyps:string):int  -insertRow(table:string, data:string):int  +connect(options:string):**connection**  +getTable(options:string):resultSet  +getUserData(options:string):resultSet  +getPHCData():resultSet  -setUserData(options:string):resultSet  +setPHCData(options:string):resultSet |