New Age Banking

Blockchain Implementation
Emmanuel Chalumeau, Githendu Mukiri
Department of Computer Science and Networking
Wentworth Institute of Technology
Boston, MA 02115, USA
{chalumeaue@wit.edu, mukirig@wit.edu }

Abstract – New Age Banking is a money sharing platform that utilizes blockchain as a means of security.

Keyword – blockchain

I. Introduction

Ever forget your wallet? Didn't have cash on you? Now, imagine you're out with friends. This must be the worst feeling ever. You want to buy something or, get into an event or party. However, you can't. You forgot your wallet, you lost your cards, or you simply don't have cash on you. The person you are with does, but what use is that to you? But wait! Imagine if you had a platform where you can share money with friends, instantly and securely without sharing your transaction information with a middle man.

II. Problem

Today, people are constantly on the go. They need quick and easy solutions to help with whatever life throws at them. systems like Venmo, Paypal, CashApp, and Zelle allow for quick money transfer between friends, regardless of what bank someone might come from. While these systems work, they are not the most secure. For example, on CashApp if someone sends you money and it is still in the CashApp portal (meaning you have not deposited the money from the app to your Bank Account) then they can request a refund and take the money back (obviously making up a lie). This is a major issue. This hole in the system, while it might seem minuscule on certain occasions, for bigger money transfers & business owners this can become a major problem very quickly.

III. Solution

For this reason, we want to look at Blockchain as a solution to this problem. **Blockchain** is a system in which data is stored/protected using cryptography and maintained across several computers that are linked in a peer-to-peer network [1]. We decided to use Blockchain, because while it is widely used for Cryptocurrency, there is a plethora of uses that have not been looked at before. In looking at the security that Blockchain provided with its hashing, we felt that it was the perfect tool to implement [2]. We made a platform that allows two users to send money to one another as well as request money from one another. We implemented Blockchain into our project to track/store our transactions. Through, the video Simple Blockchain in 5 Minutes [3], we were able to take the code provided and change it so

that it reflected the needs of a transaction id and our implementation. Our implementation - User(Figure 2) 1 enters the information needed to create its object, then User 2 does the same. From there they enter into a session. A session is when people are able to send/request money with each other for an endless amount of time until they decide to stop. The simulation then allows Users to begin sending money back and forth. First, it allows someone to enter their username as a way of saying they are the one either sending money or requesting it. Then it asks who are you working with. This is meant to declare the user that why will either be getting money from or giving money to. If they chose to send money, then once the Blockchain*(Figure 1)* approves of the transaction id then it will send the money. However, if they decide to request money, the other user must agree and then only after the approval of the Blockchain will it allow the money to be transfered. After each transaction, the simulation will ask if that is all, and if it is then the simulation will display the chain and end the program.
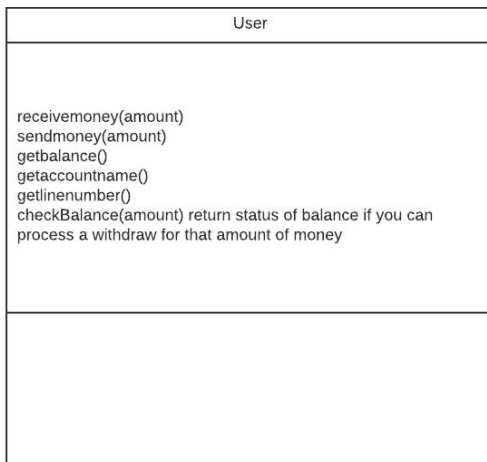
Our project implements blockchain successfully. However, before we came to this final version we had a few errors. The biggest issue that we faced was how we would keep the user's balance updated. We limited our use of classes and we decided to keep everything within the User Class and the Transaction Chain. In addition, we kept the entire simulation with in the main class. This allowed us to keep the data up to date, and ultimately allow the simulation to run more smoothly. Some of the other issues that we faced was being able to locate the users, so that data could be pulled from them. We used locator methods in the User class as well as utilized a Python **list** as a means of choosing what users would be sending money or receiving and vice versa. Overall, we were able to utilize Blockchain as a way of creating a secure money sharing platform.

Figure 1

Figure 2

```
┌─────────────────────────────────────────────┐
│                    User                      │
├─────────────────────────────────────────────┤
│                                              │
│  receivemoney(amount)                        │
│  sendmoney(amount)                           │
│  getbalance()                                │
│  getaccountname()                            │
│  getlinenumber()                             │
│  checkBalance(amount) return status of       │
│  balance if you can                          │
│  process a withdraw for that amount of money │
│                                              │
├─────────────────────────────────────────────┤
│                                              │
│                                              │
│                                              │
│                                              │
└─────────────────────────────────────────────┘
```

IV. Conclusion

We took on this topic as a way of looking at how we could make money sharing more efficient and secure. In this project, we discovered how Blockchain was used in how it could allow us to store and secure the transaction ids of transactions between users. In the end, we were able to successfully implement Blockchain into a money sharing platform.

V. References

[1]What is Blockchain?: https://www.youtube.com/watch?v=SSo_EIwHSd4

[2] Why Blockchain matters?: https://www.youtube.com/watch?v=sDNN0uH2Z3o

[3] Simple Blockchain in 5 Minutes: https://www.youtube.com/watch?v=MViBvQXQ3mM