THE UNIVERSITY OF HONG KONG
DEPARTMENT OF COMPUTER SCIENCE

Final Year Project - Intermediate Report

Stablecoin for HKD

Lo Kai Man (3035488495)
Zhou Haihui (3035470412)

Supervisor
Dr. Tsz Hon Yuen, John

January 23, 2021

Abstract

In recent years, blockchain and cryptocurrency has been popular. The market has grown exponentially. People started looking at the possibility of adopting it as a new way of electronic payment. As Bitcoin and most other Altcoins have high volatility, the market hesitates on the risk. Stablecoin has become a new variation under cryptocurrency which is pegged to some fiat currency with stable exchange rate. Tether, also known as USDT, is one of the most popular examples that has a 1:1 exchange ratio with USD. While Tether set its company in Hong Kong, there are no existing stablecoins pegged to HKD. This project will be constructed with TRON, the blockchain platform which charges zero transaction fee, to develop a new stablecoin on HKD named the Hong Kong Dollar Token(HKDT). Moreover, a crypto wallet will be developed in React Native at the same time to provide cross-platform experience for non-technical users on peer-to-peer HKDT transactions. Besides the basic transaction function, additional features such as blacklist on dishonest spending will be researched. At the time of this report, the basic prototype has been developed and is waiting for testing to evaluate the performance. Utility features such as QR code generation and scanning is provided. In the next phase of development, more advanced features such as blacklist will be provided.

Acknowledgement

First, I would like to extend my deepest appreciation to the Department of Computer Science, and Dr John Yuen for offering this precious chance on this project. Without Dr John Yuen's invaluable advice, this project cannot be on the right track.

Last, I am extremely grateful to Miss Mable Choi for her guidance and suggestion when writing this report.

List of Figures

2.1 Rendering in React Native

2.2 Flow of Smart Contract Blockchain

2.3 Example of Smart Contract Transaction in TRON

3.1 Logic Flow Chart of Buying Scenario

3.2 TRC-20 Standard

List of Tables

3.1 Backend Design

3.2 Project Release Plan

3.3 Project Timeline

List of Abbreviations

HKD	Hong Kong Dollar
HKDT	Hong Kong Dollar Token
iOS	iPhone Operating System
QR Code	Quick Response Code
ReactJS	React.js
TPS	Transactions per Second
TRX	Tronix
UI	User Interface
USD	United States Dollar
USDT	United States Dollar Token

Abstract 2

Acknowledgement 3

List of Tables 4

List of Abbreviations 4

1. Introduction 7
    1.1 Overview of Cryptocurrency 7
    1.2 Stablecoin 8
    1.3 Smart Contract 8
    1.4 Crypto Wallet 9
    1.5 Payment Method 9
    1.6 Objectives 10
    1.7 Project Contribution 10
    1.8 Outline of the report 10

2. Methodology 12
    2.1 Introduction 12
    2.2 The Development Framework: React Native 12
    2.3 The Blockchain Platform: TRON 13
        2.3.1. Introduction 13
        2.3.2. Transaction Fee in Blockchain 13
        2.3.3. Bandwidth System in TRON 14
        2.3.4. Energy System in TRON 14
        2.3.5. Freeze Balance in TRON 16
        2.3.6. Shasta Testnet in TRON 16
    2.4 Summary 16

3. Current Progress 17
    3.1 Overview 17
    3.2 Application Side 17
        3.2.1 Transaction Flow 17
        3.2.2 Protecting the Private Key 19
    3.3 Backend Library 19
    3.4 Smart Contract Side 21
        3.4.1 Smart Contract under TRC-20 Standard 21
        3.4.2 Blacklist on Dishonest Spending 22
    3.5 Testing 22
    3.6 Project Schedule 22
    3.7 Future Plans 24
    3.8 Challenges 25

3.8.1. Dishonest Spending 25
3.8.2. Transaction Fee For Zero Energy 25
3.4.3. Address Registration Fee 26
3.9 Summary 26
4. Conclusions **27**
Appendix A **28**
Appendix B **30**
Appendix C **31**
References **32**

1. Introduction

1.1 Overview of Cryptocurrency

Cryptocurrency is the first application of blockchain and one of the successful examples of digitalized assets. Ever since Nakamoto[1] issued the first Bitcoin, its total value has risen by over US$657 billion by January 2021 [2]. For the cryptocurrency market[2], there are more than 8,257 cryptocurrencies in the market and carrying a total market value over US$1.01 trillion. There are at least 2.9 million users actively involved in cryptocurrency trading[3]. All these figures indicate an active market that has been growing rapidly and is full of potentials.

Many research studies have summarised the main features and advantages of cryptocurrency like Bitcoin. It is decentralised, transparent and flexible[4]. Conventional currencies would usually have a centralized party overseeing the whole currency system and step in when needed, while Bitcoin would be peer-to-peer. Bitcoin broadcasts transactions to every miner, who contributes their processing power to verify a transaction. The algorithms are open-sourced as well. An account can be generated mathematically. The transaction address and authentication process are then handled with asymmetric encryption with the account as cryptographic keys. Therefore, everyone will have access to the records and integrity is ensured by algorithm and maths. In addition, transactions happen fast with low fees[4]. In fact, as there are no authority to handle transaction verification, the transaction fee is actually paid to a special role in cryptocurrency, the miners. The action of mining is referring to the heavy amount of computation work for validation and appending new transactions to the blockchain network. The transaction fee is then provided as the incentives for miners. The transaction fee for Bitcoin is around US$2 for each transaction in October 2020[5].

People may purchase cryptocurrency for investment only while some shops accept bitcoin for buying goods or services. For example, Bitcoin is accepted as a payment method on overstock.com[6], an American e-commerce retailer, or digital contents from Microsoft[7]. Another research[3] shows that 52% of the company participants in the study provide service for processing cryptocurrency payment with merchants.

However, Bitcoin has not been accepted in the mainstream finance system or as a decentralized payment method due to its high volatility. The price of Bitcoin fluctuates over a wide range from day to day. The Bitcoin dropped to around US$5,000 in March 2020 and the current price has reached US$35,000. This is the main reason for most of the merchants accepting Bitcoin as payment, they would exchange it immediately to fiat currency once received. The volatility undermines cryptocurrencies' potential to be used in online payment and therefore a new variation of cryptocurrency was proposed, the stablecoin.

1.2 Stablecoin

As its name, stablecoins are stabilised cryptocurrency with constant exchange rate. They are pegged to fiat assets or fiat currencies, but can be using different collaterals. The mechanism is classified to four types[8].

The first type, fiat-collateralized stablecoin is collateralized to fiat currencies such as US dollars or pounds. The unavoidable disadvantage is that a centralized custodian has to be introduced and make sure the exchange can be made at a fixed rate. One famous example is Tether, the USDT, having the total market value of US$15 billion[2] is the biggest stablecoin in the market. The exchange rate stays at 1 USDT to 1 USD.

The second type, commodity-collateralized stablecoin, is very similar to the first type. They use different collaterals of commodity types such as gold or oil and share a similar disadvantage with the previous type.

The third type is crypto-collateralized. It uses other cryptocurrencies as collaterals. Anonymous custodians is possible under this setting. The Dai[9] is an example for crypto-collateralized stablecoin, that is pegged to 1:1 USD, but users are using Ether, the second popular cryptocurrency[2], as a deposit.

The last type is a non-collateralized stablecoin. That is, they stabilise the purchasing power algorithmically. For example, when the purchasing power increases, the balance or payment amount are adjusted correspondingly that users pay at a lower price for the product[10]. However, there are no existing mechanisms that could ensure both the stability and decentralization[8].

Despite all the challenges, a stablecoin still has the potential to be a fast and secure payment alternative. However, there are no existing stablecoins that are directly pegged to HKD. When one has the incentive to issue a new cryptocurrency, one needs to at least implement a smart contract.

1.3 Smart Contract

Smart contract is a predefined set of logics and rules. It is implemented as a script that executes automatically when received a message or transaction as input[11][12]. Transactions can then be carried out peer-to-peer. For example, a smart contract may have a “transfer” function that will send X units of crypto coins to the receiver and deduct the same amount from the sender.

Moreover, it has the following characteristics. Firstly, a smart contract should be properly designed to handle all possible conditions[11]. For example, the transfer action should not be

made when the sender does not have the amount to pay. Secondly, the output should always be the same when given the same input[11]. As in blockchain, it uses a consensus rule that only the execution result is agreed by the majority of the blockchain network. If the smart contract is non-deterministic and gives unpredictable results for each run, it is impossible to reach a consensus when the transaction is broadcast to all the nodes on the network. Thirdly, a smart contract is deployed on the blockchain[11][12]. In this distributed way, every internet user can access the content. Fourthly, all actions taken by the smart contract are irreversible[12][13]. One cannot roll back any transaction after it is sent to the smart contract.

There are a few popular platforms and standards that help developers to easily set up and establish their cryptocurrency to the blockchain network. This study is going to use TRON. A more detailed analysis will be provided in the methodology chapter. For users with no knowledge in blockchain or smart contract, a crypto wallet is required as an interface for managing their cryptocurrencies.

1.4 Crypto Wallet

Crypto wallet is the application that handles buy, sell and transfer of the cryptocurrency. The provided features differ by needs.

When building a crypto wallet, the most important consideration is about securely storing the private key[14]. As in blockchain networks, the private key is used to generate signatures and authenticate all operations. The key must be kept secret. Otherwise, malicious users can access the digit assets stored in the corresponding address and all the lost cannot be reverted. In addition, if the user loses the private key, he or she loses the access to the wallet. Some wallets have wallet seeds for recovery. That is a set of vocabularies or alphanumeric for generating the private key. Compared to private key samples in Ethereum, which is 64 hex characters in the form like “0xFF00…00FF00FF00”, wallet seeds are relatively easier to remember and therefore can be used to recover the private key. Consequently, protecting the wallet seed is as important as protecting the private key.

With use of the crypto wallet, even the users have no knowledge about the blockchain, they can use a set of simple controls to handle the trading of cryptocurrencies. The users of the crypto wallet only need to know how to use the UI of the wallet provided.

1.5 Payment Method

Hong Kong has a lot of payment methods, such as Cash, Octopus Card, Credit Card or FPS. Compared with those in operate payment methods, HKDT payment, or stablecoin payment has outstanding behaviors. Even stablecoin payment requires a trusted third party to issue the stablecoin, transaction of stablecoin payment work likes the Cash payment where the transaction

will not involve any third party. In other words, third parties can be ignored if the user does not require a deposit or withdrawal.

Moreover, stablecoin will do and store all the transactions in the distributed blockchain system. Thus, the trusted third party cannot know what transaction you have involved apart from the one that you used to receive the stablecoin sent from the third party. Also, the transaction recorded in the blockchain system can be no dispute and this unique behaviour can ensure the data integrity [29].

1.6 Objectives

Since stablecoin payment has a potential market, this project aims to establish the stablecoin for Hong Kong Dollar (HKD) that is called Hong Kong Dollar Token (HKDT) for promoting the HKDT payment in Hong Kong. This project aims at solving two technical parts including building the smart contract of HKDT on the blockchain and building an user-friendly mobile application for non-blockchain users to enjoy the HKDT services.

1.7 Project Contribution

Hong Kong, as an international financial centre, has a unique advantage in promoting the usage of cryptocurrencies. In fact, Tether, the issuer of the most popular stablecoin USDT, set its company in Hong Kong[15].

Although there are not specific regulatory measures on cryptocurrencies, Hong Kong Monetary Authority (HKMA) and Securities and Future Commission (SFC) have recognised Bitcoin as a virtual commodity. Other crypto coins are likely to fall into similar aspects[15]. Currently, transfer of cryptocurrencies between individuals is allowed as far as it does not involve money laundering activities. No license is required for trading or owning cryptocurrencies. Moreover, the government has planned to encourage financial innovation in the city. For instance, Hong Kong has approved a cryptocurrency fund for the first time on 20th April 2020[24]. Although with uncertainties on future regulations, the legal and social environment is friendly to embrace blockchain applications.

Studying the feasibility of having a stablecoin in HKD aligns with the future trend of fintech. It turns cryptocurrency into a topic that is related to everyone. Although the legal environment has not recognised cryptocurrency as electronic money, the potential has to be highlighted.

1.8 Outline of the report

The report contains four chapters. The first chapter introduces the motivations for building a stablecoin in HKD and classifies the existing stablecoin. Moreover, it explains the technology and concepts that are required to build a new cryptocurrency and illustrates the goal and significance of this project.

Chapter two examines the methodology. That includes the two major frameworks adopted for the crypto wallet side and smart contract side, and the advantages brought to this project.

Chapter three describes the current progress, which includes system design and techniques. It defines two focuses when testing the deliverable. Project timeline and next steps are covered and limitations and challenges are analysed.

Chapter four concludes the report. It summarises the progress and future works presented in Chapter three.

2. Methodology

2.1 Introduction

In this chapter, two important methodologies will be introduced. First is React Native, which is the application framework that will be used to build the crypto wallet. Its performance and capability could profoundly affect the user experience with HKDT. Second is TRON, which has the potential to make free transactions possible. It is a growing blockchain platform but with potential to compete with the more famous Ethereum and Bitcoin.

2.2 The Development Framework: React Native

React Native is a cross-platform mobile application framework. It was released by Facebook in 2015[17]. Before React Native, if a developer wants to establish a mobile application on the two most common platforms Android and iOS, the developer needs to write two sets of code respectively in Java and Swift. Moreover, extra effort is required to ensure both versions have the same view and behaviour for consistent user experience. The main idea of React Native is to allow developers to only maintain one piece of code for multiple platforms[17].

It is noteworthy that React Native is built in a similar technical structure as ReactJS, a Javascript framework widely used in web development also created by Facebook. That implies the potential for code developed with React Native be reused on web application. Having a crypto wallet that is available for devices of any platforms will help users to adopt the new payment method, this cross-platform ability is highly valued.

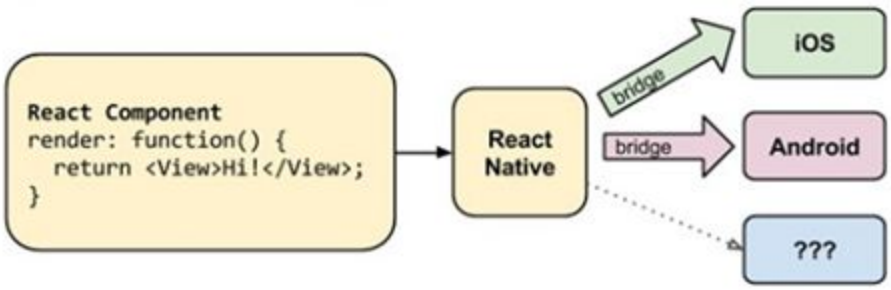


Figure 2.1: Rendering in React Native

Source: Adopted from [18]

Figure 2.1 demonstrates the internal structure of a React Native application. The React Native side handles the structure, the logic and functionalities of the application. The native side is responsible for visualisation and user gestures. They run on two separate threads with an

asynchronous bridge between them for communication. For example, when the user opens the application, the React Native is first activated and informs the native side of how the elements should be displayed. When the user presses a button, the native side will send this message to the React Native side and the application will decide what to do based on the logic flow in the code implementation. The React Native will automatically determine what native function it should call and what native type of data should be sent. This structure allows instant refresh and easy event handling[18][19]. In this way, the React Native application can be easily migrated to different platforms.

In addition, research[19] shows React Native application and native application have similar performance in terms of memory usage, frames per second measurements, response time and size. Furthermore, React Native resembles almost identical user experience. As React Native is open-sourced and the community has been actively updating and contributing to new features, it has great potential to look for in the future.

2.3 The Blockchain Platform: TRON

2.3.1. Introduction

TRON is a decentralized blockchain platform established in March, 2014[20]. It is one of the popular blockchain systems in the world. The total market value ranked 20th in all existing cryptocurrencies[2]. It is also believed to be the second most popular token development platform after Ethereum[28]. TRON is adopting the smart contract standard compatible with Ethereum as well to attract developers. The main difference between TRON and Ethereum, the more popular blockchain platform, is that TRON adopts another unique mechanism to make zero transaction fee possible.

2.3.2. Transaction Fee in Blockchain

The developed objective is promoting the Decentralized Applications (HKDT smart contracts), the alias of smart contract in TRON, in the TRON ecosystem, so TRON applied the concept of Bandwidth. Before introducing the Bandwidth system in TRON, one should know that every blockchain user needs to pay transaction fees whenever they create a new transaction. The transaction fee can prevent some adversaries broadcasting lots of useless transactions to the blockchain system and cause Distributed Denial-Of-Service (DDOS) attack problems because the miners need to solve the cryptogram puzzles so that other miners can confirm the new block reward is belonged to the corresponding miner. Therefore, the number of transactions can be confirmed is limited by the number of blocks can be confirmed in one period. If there is overflow of useless transactions in the transaction pool, the real transaction is less likely to be picked up by the miner to create blocks. This essential measurement results in even if someone is using the smart contract, he or she has to pay the transaction fee in the blockchain. For example, if there is

one smart contract with token AToken in the blockchain system with coin BCoin, whenever users want to trade 1 AToken to other users, they may need to pay some BCoin as the transaction fees. That is not good news for stablecoin. According to the example in the Fig. 2.1, if there is a HKDT smart contract in blockchain system Ethereum with a coin named Ether, users need to hold some Ether so that when they want to trade HKDT, they have some Ether to pay for the transaction fee. The contradiction is that users want to hold some stable cryptocurrencies, but at the same time they must hold some unstable cryptocurrencies so that they can do the trading.

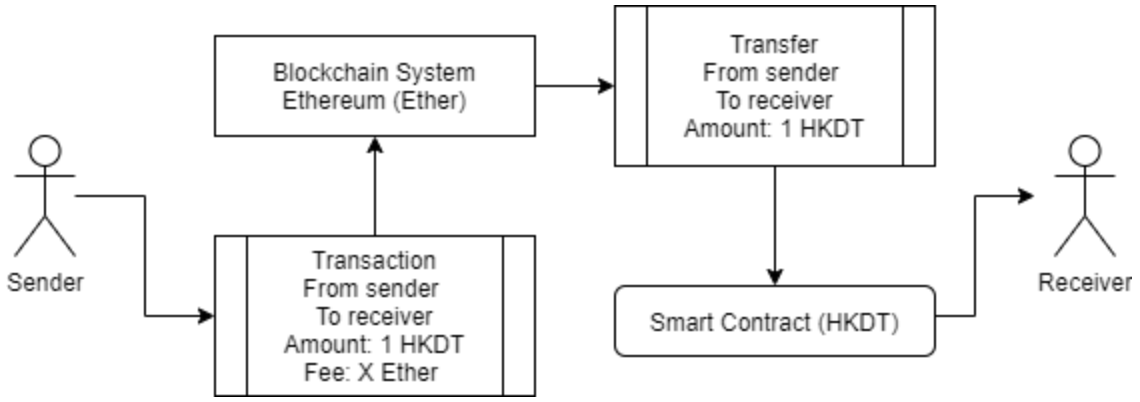


Fig. 2.2 Flow of Smart Contract Blockchain

2.3.3. Bandwidth System in TRON

The Bandwidth System can alleviate the situation of stablecoin by providing a transaction fee pool. Each user has a Bandwidth Pool with free 5000 bandwidth per day and the bandwidth in the Bandwidth Pool is used as the transaction fee [27]. Users need to buy the bandwidth when all the free bandwidth is used to pay for the transaction fee instead of directly paying the transaction fee. Since a normal transaction required around 200 to 300 bandwidth, everyday the users can enjoy around 20 zero transaction fee trading. That means stablecoin users using the HKDT smart contracts in TRON can prevent the paying of the transaction fee. They can create several addresses, that is the alias of an account in the blockchain, and get 5000 free bandwidth per day in every address. If a stablecoin holder has 10 addresses in TRON that means he or she can have around 200 zero transaction fee trading per day. Of course, to prevent the DDOS attack mentioned before, creating a new address required 1 TRX, the coin in TRON, as the registration fee. If someone is willing to pay enough TRX to register lots of accounts, DDOS attack is still possible but they cannot earn from the attack, so it can be safe while TRON users can earn the registration fee from doing enough zero transaction fee trading. To conclude, TRON increased the possibility of DDOS attack, but the benefit of zero transaction fee is decisive for applying HKDT in TRON.

2.3.4. Energy System in TRON

Apart from the bandwidth system, another resource, the Energy System, is important to the HKDT smart contracts [27]. If bandwidth is treated as the transaction fee, the energy system is

monitoring measurement. HKDT smart contracts have a high degree of freedom because the HKDT smart contracts only need to ensure it is using valid Solidity, a programming language for smart contracts. Thus, the contract may have loopholes that may crush the blockchain platform such as the most popular issue, unbounded looping. For example, there is one loop in the contract used to check the total supply handed over.

```
getSupply():
    Init supply as 0
    For each address in the address_list:
        supply = supply + the balance in address
    return supply
```

According to the coding above, it gets the total supply handed over by sum up the balance in each address. The program is valid, but if there are 1 millions addresses in the address_list and there are thousands of users calling this function frequently, it definitely increases the workload of the blockchain provider because users are willing to get information from the blockchain provider which is more credible. Thus, the Energy System is proposed for HKDT smart contracts to prevent users using the high workload HKDT smart contracts freely. With the Energy System, according to the Fig. 2.3, every HKDT smart contracts transaction is required to pay both bandwidth and energy as the transaction fee if the transactions are valid. However, if the HKDT smart contracts transaction is invalid, such as timeover, it will use up all the energy stored as a penalty.

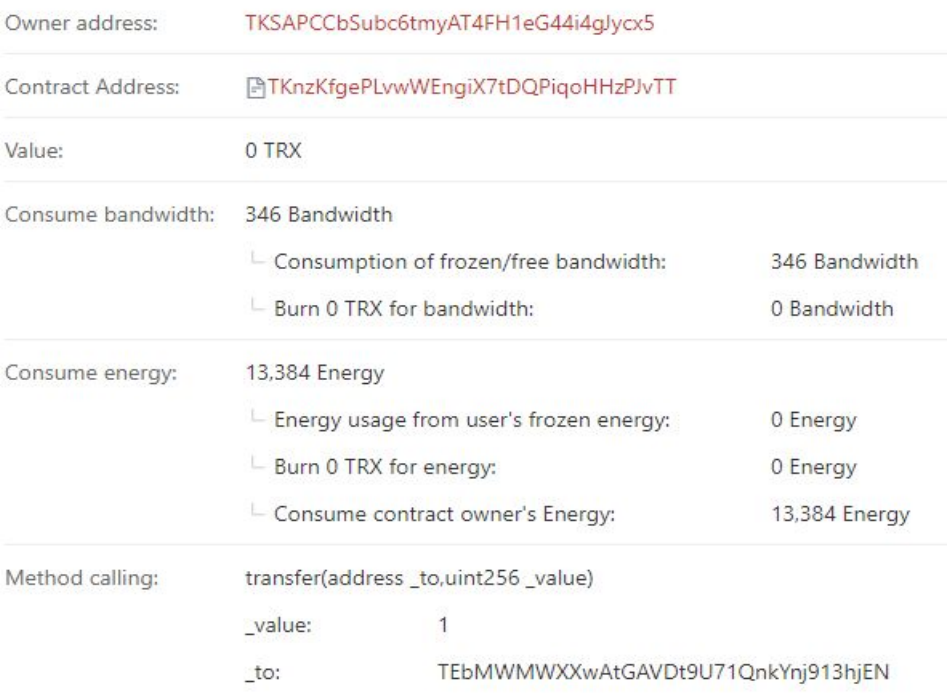


Fig. 2.3 Example of Smart Contract Transaction in TRON.

2.3.5. Freeze Balance in TRON

Since HKDT smart contracts transactions require energy which is not freely given as the bandwidth, freeze balance service in TRON can be used to earn the free energy for achieving zero fee HKDT smart contracts transactions. Freeze balance service is freeze TRX, the name of the Coin in TRON, to get energy while the energy obtained from the service is equal to the number of TRX frozen divided by the total amount of frozen TRX from different users times 50000000000 [27].

```
energy obtained = user's TRX frozen amount / total amount of frozen TRX in TRON * 50_000_000_000
```

The total amount of energy is shared between all the users in TRON, so the more the TRX is frozen, the more free energy can be obtained. Since the energy obtained can be automatically charged within 24 hours by the way of reducing the consumed energy. For example, Amy used 24 energy and there is no consumed or obtained in the coming 24 hours. Then, the consumed energy becomes 23 after one hours, no matter how much energy Amy has. Therefore, the HKDT smart contracts can achieve zero transaction fee by using the free bandwidth and free energy obtained from frozen TRX in the contract. According to the example in Fig. 2.3, the energy consumption can be fully beared by the HKDT smart contracts, in such a situation, the user can ignore the Energy System to enjoy the zero fee transactions.

2.3.6. Shasta Testnet in TRON

Although zero fee HKDT smart contracts trading can be achieved in TRON, this project is not going to use the Mainnet in TRON because of some compulsory payment such as registration fee, which is explained in the challenges. Every blockchain system would have multiple subnets and there must be one Testnet provided for users who want to test the reliability of corresponding blockchain systems or developers who want to test the HKDT smart contracts before publishing to the other subnets. Testnet is just totally copied from the mainnet, but they will have different blocks and chains. The unique feature provided in Testnet is providing free faucet services. In TRON, that means, all TRON users in Shasta Testnet can get free TRX, the coin in TRON, to do the trading. With the use of Shasta Testnet, this project can test the HKDT smart contracts freely and different situations that may be faced after publishing the HKDT smart contracts in the Mainnet.

2.4 Summary

This chapter has explained the two major frameworks and their potential in building a stablecoin for HKD. Their working principles and their significance to the project were described. The next chapter will be the current progress.

3. Current Progress

3.1 Overview

This chapter will demonstrate what has been done for this project. System implementation of the application and the smart contract will be explained. After illustrating how the first prototype will be tested, it will also mention the project schedule and the next steps. Moreover, limitations and challenges will be covered.

3.2 Application Side

This section will describe the progress on the application side. The UI design and the screen capture of real implementation are appended as the Appendix A and Appendix B. They will not be explained in detail. This section will focus on technical content and start with illustrating a transaction scenario. Then, it will explain other measures implemented for the transaction with HKDT to be possible and secure.

3.2.1 Transaction Flow

Before any application is built, we need to outline the logic flow when using the HKDT with the HKDT Wallet for transaction.

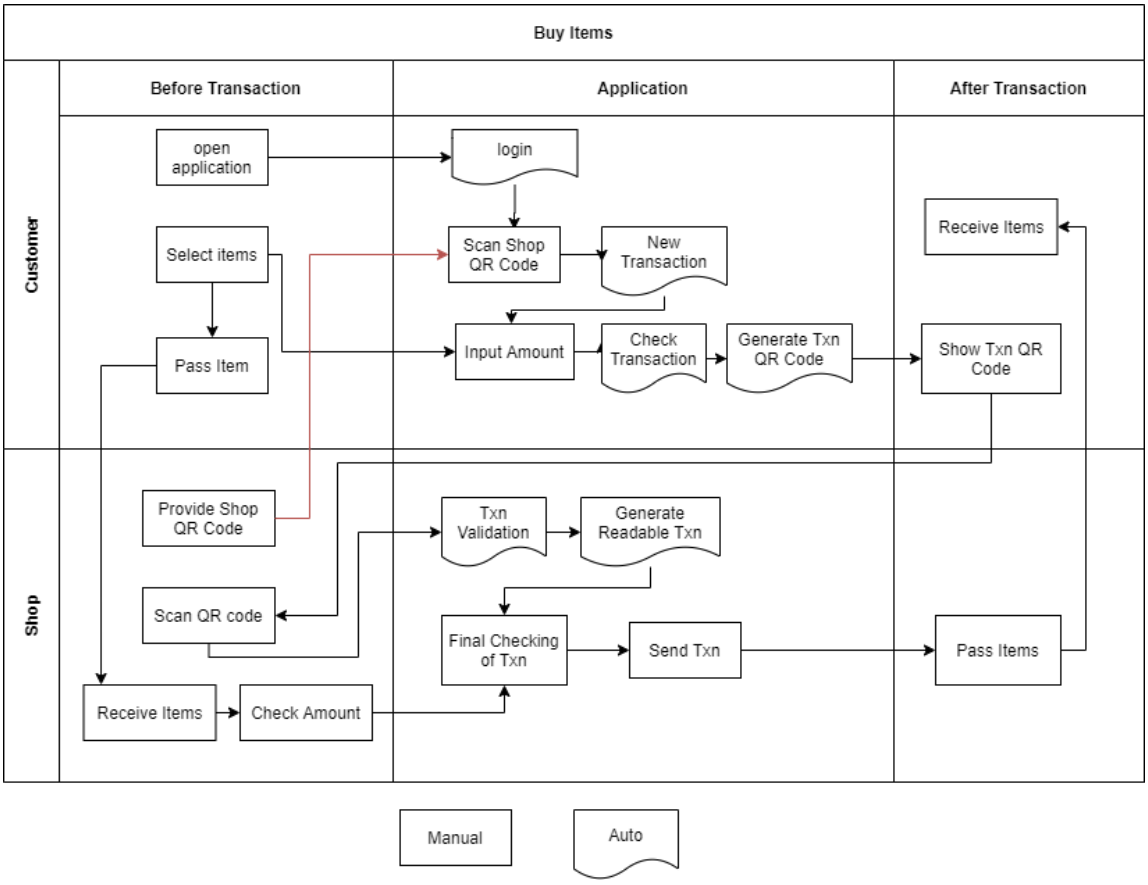


Figure 3.1 Logic Flow Chart of Buying Scenario

The flow chart (Figure 3.1) demonstrates the logic flow when a HKDT transaction takes place between a buyer and seller. The whole process begins with the shop showing its address encoded in the QR code to the customer. Then, the customer should unlock the application and select the account he or she would like to pay for the transaction. After scanning the QR code, a new transaction will be created waiting for the user to input the amount. The transaction is then digitally signed and generates a QR code. The shop receives the signed transaction through scanning the QR code. The seller can validate the transaction from their side and generate a receipt before sending it to the blockchain network.

It might appear strange that a single transaction involves scanning QR code twice but the intention is to protect the seller. In the crypto world, the seller only receives the amount to the account after the transaction was broadcasted and verified by the blockchain network. Due to network delay, if the buyer decides to disconnect the device from the internet after creating the transaction, the transaction will never be sent out and the receiver will never receive the amount. As a transaction message must be signed by the account owner who is giving out money while the broadcast step can be carried out by any individuals, having the seller to send out the transaction is in their best interest. The buyer should digitally sign the transaction same as

signing a cheque and the seller is advised to broadcast it as soon as they receive the signed transaction.

This mechanism protects sellers and it shortens the time for transaction as both parties do not need to wait until the transaction gets through the blockchain network and gets verified, which can take as long as 1 minute[25]. While the seller will make sure they send out the signed transaction, the buyer can leave the counter after the balance is verified to be enough.

3.2.2 Protecting the Private Key

Private key is the only identifier for all authentication in blockchain. It must be kept only on the local device and even the application should not store the key in plain text. Symmetric encryption is implemented here.

Symmetric encryption refers to an encryption method that uses the same key for encryption and decryption. With symmetric encryption, when the user imports or creates an account, the private key is not stored in plain text but encrypted text. The user will be asked to input a password of their choice to encrypt the private key and they will be prompted to decrypt it whenever they want to view the details such as transaction history of an account or signing a transaction.

In this way, the application shall not be the fragile point in protecting the private key. However, the user should still stay mindful in keeping their private key and the password only to themselves.

3.3 Backend Library

The first version of the application library has been developed. It includes basic functions that support transactions with HKDT.

We define the responsibilities of the library as three parts. The first part is interacting with the smart contract. This part involves calling methods defined in the smart contract. The second part is interacting with the blockchain platform. That includes fetching data from the internet related to the blockchain such as energy resources balance and transaction history. The third part is connecting the backend to the mobile application. The application side should be able to define all necessities from the backend with minimal setup.

Table 3.1 Backend Design

Function(Part 1)	Input	Output

constructor(setup)	setup : parameters that including network setting and smart contract address	/
init(txn)	txn : parameters for initializing transactions, including but not limited to sender address, receiver address, amount and transaction type.	/
getHash()	/	Return the hash value of the initialized transaction.
checkBalance(address, amt)	address: the account address amt: the amount required	Return a Boolean on if the address has enough HKDT
serializaTxn()	/	Return the serialized version of the transaction.
sendTransaction()	/	Return the receipt of broadcasted transaction.
getConfirmation(txnId)	txnId : the transaction id	Return the number of confirmations of the transaction.
checkAddress(address)	address: the account address	Return if the address is valid.
getAddress(pubKey)	pubKey : public key	Return the address calculated from the public key.
signTransaction(privateKey)	privateKey : private key for signing the transaction	/
Function(Part 2)	Input	Output
getBandwidth(address)	Address: the account address for checking bandwidth	Return the bandwidth number in decimal
getEnergy()	/	Return the energy balance of the contract in decimal
getBalance(address)	address : the account address	Return the balance of HKDT in the input address.
getTransactions(address)	address : the account address	Return the transaction including trc-20 tokens related to the address

Table 3.1 has displayed the library structure. The function column defines the function name with input parameters inside the brackets. The function and input parameters are names used in the original code, therefore they are following lower camel case format and traditional short forms. The names are not guaranteed to be grammatically correct. The functionality of each library function is clearly defined and separated by its inputs and outputs.

The backend library is implemented to assist those features provided in the application.

3.4 Smart Contract Side

Smart contracts fundamentally define what the cryptocurrency is capable of. It has to follow some common standard so that every program running on a particular blockchain network could understand it. The TRC-20 standard has to be well-tested and self-explanatory as it will be public. Above all, a published smart contract is permanent and cannot be altered, that is the property makes a smart contract trustworthy. Therefore, every feature must be decided with extra attention. There is one feature that has been determined as essential to be offered which is the blacklist function.

3.4.1 Smart Contract under TRC-20 Standard

Since the smart contract for HKDT will be deployed with TRON, the smart contract will follow the TRC-20 standard.

```
contract TRC20Interface {
    function totalSupply() public constant returns (uint);
    function balanceOf(address tokenOwner) public constant returns (uint
balance);
    function allowance(address tokenOwner, address spender) public constant
returns (uint remaining);
    function transfer(address to, uint tokens) public returns (bool success);
    function approve(address spender, uint tokens) public returns (bool
success);
    function transferFrom(address from, address to, uint tokens) public
returns (bool success);

    event Transfer(address indexed from, address indexed to, uint tokens);
    event Approval(address indexed tokenOwner, address indexed spender, uint
tokens);
}
```

Figure 3.2 TRC-20 Standard

Source: Adopted from [21]

The basic form of a smart contract deployed with TRON, as shown in Figure 3.2, contains six functions. The implementation will be written in TRON Solidity, the programming language for TRON. Fundamentally, the smart contract needs to be able to perform action such as telling the balance of one's account and transferring tokens. The HKDT will contain all the functions although the implementation is still waiting for more testing. Additional features can be added to the contract when necessary.

3.4.2 Blacklist on Dishonest Spending

A fatal flaw on decentralised payment is that most verifications have to be done through the network. That leaves room for dishonest spending.

The HKDT will adopt a similar implementation of blacklist in USDT as the smart contract shown in Appendix C. The blacklist is embedded in the smart contract as an abstract state. The list is only allowed updating from an address designated as the owner. Therefore, only the owner with the private key is able to command the contract to append or remove the blacklist state accordingly.

3.5 Testing

Testing needs to be properly designed beforehand to make sure the application will be evaluated correctly. Testing on the first prototype is not yet performed. This section is to define the testing content.

The first focus is measuring the average transaction time, energy consumption and bandwidth consumption. These are the main factors that will affect users' acceptance level on adopting HKDT as a transaction method. As the performance in real life will be affected by network status, twenty testing transactions will be measured to give a rough evaluation on the HKDT wallet.

The second focus is testing the ability in exception handling. The wallet application has to be resilient enough to handle exceptional events such as invalid inputs from the user and timeout due to an unstable network. Unit tests should be placed, where the outcomes from the application are checked to see if they match with the expected outcomes. For example, if the input address is invalid, the application should be able to detect that and prompts the user to re-enter it.

Sufficient testing is part of the result evaluation and it helps to make sure the project is on the right direction with the project objectives.

3.6 Project Schedule

For better project management purposes, the release of each version and the timeline are planned.

Table 3.2 Project Release Plan

Technical Focus	Versions	Features
Smart Contract	Version 1	Basic TRC-20 Contract
	Version 2	Transaction Fee Handling and Blacklist Function
Application	Version 0	UI Prototype
	Version 1	UI Implementation Transfer Function Using Address
	Version 2	QR Code for Transaction
	Version 3	Communicate with NodeJS Server
	Version 4	Handling Transaction Fee and Blacklist
NodeJS Library	Version 1	Provide Related Function for HKDT Transaction
	Version 2	API to NodeJS Server
NodeJS Server	Version 1	Transaction Validation and Backup
	Version 2	Blacklist Handling
	Version 3	Simulate HKDT and HKD Exchange

Table 3.3 Project Timeline

	Smart Contract	Application	Library	Server
Oct	Version 1	Version 0		
Nov		Version 1	Version 1	
Dec				
Jan		Version 2		Version 1
Feb	Version 2	Version 3		Version 2
Mar		Version 4	Version 2	Version 3

Apr				

Currently, the design of the application interface is finished and the basic version of smart contract has been drafted as stated in Table 3.1. The version 1 library has been implemented in the structure shown in section 3.2.2 and the implemented application view can be found in Appendix B.

According to Table 3.3, this project will host a simple smart contract in the blockchain system for testing and designing a prototype of mobile application including the UI as well as the functionality in October.

Then, this project will use the prototype to build the react native mobile application and establish a library for the mobile application to connect to the blockchain system. Thus, this project will be able to use the mobile application to finish the smart contract transaction in November.

In January, this project will implement the QR code functionality in mobile applications for peer-to-peer offline communication between mobile applications and start to set up the server with simple transaction backup and validation functions.

In February, this project will host the new smart contract in the blockchain system to provide the transaction fee handling and backlist handling in the blockchain system. The library will be updated for providing the API from the server built in January and the mobile application will apply the service provided by the server including the transaction validation and transaction backup.

In March, the mobile application will apply the service provided by the new Smart Contract including the transaction fee handling and blacklist handling. Until the end of March, this project will be almost complete.

3.7 Future Plans

The schedule is still on the track of the planning, so a simple mobile application and a NodeJS library are built that can be used to finish the smart contract transaction in the current stage. Also, the QR code function in the mobile application is done to provide the peer-to-peer offline communication. According to the planning in Table 3.3, the work in the next stage will be building the NodeJS server as well as constructing the blacklist handling smart contract. When these two parts are builded and tested to be fine, the mobile application and the server will both then apply the blacklist functionality. After the blacklist handling stage, the remaining tasks will be providing more functionalities to solve some difficult challenges in this project and optimize the components finished in the previous stages.

3.8 Challenges

3.8.1. Dishonest Spending

The transaction time is one of the flaws that can be exploited by some offenders. According to Fig. 3.1, starting from the customer generated the Txn QR Code until the transaction being confirmed on the blockchain, the customer can send another transaction to the blockchain and use up all their assets in the new transaction. Then, if the new transaction that used up all the assets is confirmed before the old transaction, the old one may be seen as invalid because all the assets should be used up and confirmed in the last transaction. This fakement is well known as double spending. The mechanism of double spending is to try to guess the new transaction can be confirmed first and make the old valid transaction being invalid so that the receiver in the old transaction believes the old transaction will be confirmed on the blockchain.

Currently there is no solution that can solve the dishonest spending, but a blacklist handling is currently proposed to reduce the influences bringed out. The blacklist handling is adding the blacklist in the HKDT smart contracts and using the owner address of the HKDT smart contracts to control the blacklist. All the transactions done on the mobile application proved in the project will send a copy to the NodeJS server before broadcast to the blockchain. When there is dishonest spending being detected or reported, the dishonest address will be added to the HKDT smart contracts. With the use of blacklist in the HKDT smart contracts, both mobile application level and HKDT smart contracts level will reject the transaction including the addresses in the blacklist.

3.8.2. Transaction Fee For Zero Energy

Dishonest spending problems can be detected easily because evidence can be found if the transaction is doing with the mobile application. The loophole of transaction fees is difficult to be detected. In the project planning, all the energy required for DApps transaction is beared by the DApps and the bandwidth is beared by the DApps transaction sender address. Since the energy required specific frozen TRX to achieve zero transaction fee, the free energy provided in one period is limited. Although this project will try to evaluate the amount of frozen TRX that can satisfy a specific amount of users to do zero fee transactions, there are some situations such as some rascals send a lot of zero free transactions and use up all the energy out of the evaluation. Since all the energy is beared by the DApps, lots of transaction fees need to be paid by the DApps because users can still do the zero transaction fee trading. For example, 0.13 TRX (0.026 HKD) is required as the transaction fee paid by the DApps for every transaction when the energy is used up according to Table 2 and Table 3.

The current suggested solution in this project is drawing the transaction fee from the users when users try to do the transaction after the energy is used up. Since all the energy consumption

should be beared by DApps of this project is the rule deployed when creating the DApps, the plan of receiving transaction fee will be implemented in DApps.

Function in simple smart contract	Function proposed by this project
transfer(to, value): Send <value> of Token to <to>	transfer(to, value, fee): Send <value-fee> of Token to <to> Send <fee> of Token to <owner> when the fee is greater than zero

According to the function proposed by this project, the transaction fee can be obtained when the fee parameter is greater than zero value. The input param of transaction fee in the function will be checked and deduced in the mobile application level. The mobile application would have a Energy Pool to notify the users how many zero fee transactions can be done in the DApps currently and a Bandwidth Pool to notify the users how many zero fee transactions can be done in this address currently. Once the pool is being drought, specific transaction fees in unit HKD will be calculated as the parameter in the function call of DApps.

3.4.3. Address Registration Fee

The issue mentioned about sending a lot of zero free transactions has been solved by the TRON with the address registration fee. For any address that does not have any transaction inside will be seen as an inactive address, any sender wanting to do transactions with these inactive addresses will need to pay 0.1 TRX as registration fee to prevent others creating lots of addresses freely and enjoy the zero fee transaction. To handle the address registration fee, two approaches are considered.

First is adding one Activate Button to request the address registration before this address can be used for transaction. Second is to reduce the amount of HKDT that can be gained from the sender. For example, when Amy has an address that needs to be registered, Amy may be able to get 0.98 HKDT from Bob if Amy gives 1 HKD to Bob for trading. There is 0.02 HKD that is preserved by Bob in this transaction if it is assumed that the exchange ratio of TRX : HKD is 1:0.2 and 0.1 TRX registration fee is 0.02 HKD. The solution has not been confirmed and still need more research or testing to find out the better one.

3.9 Summary

This chapter provided an overview on the project progress. It started by presenting the design we have made and the justification on each design decision. It explained the transaction flow that is

different from traditional payment and the extra encryption step for protecting the private key. It also explained the structure design on the smart contract and how the blacklist feature benefits the users. Although the prototype has not been tested, the testing focuses were discussed. It further described the project plan and the next focus on the application testing. The problem of double-spending was highlighted as it could damage the credibility of HKDT.

4. Conclusions

Cryptocurrency has been growing fast over the years. The market is trying to embrace cryptocurrency as a potential next generation electronic payment. Stablecoin is then suggested and implemented in various forms. Although the Hong Kong government and legal environment favour innovation in blockchain, there has not been any stablecoin pegged with HKD being proposed. This project targets to implement a HKDT, the stablecoin for HKD, and test its feasibility. To achieve the goal, mature frameworks such as React Native and TRON are adopted, but there are still problems to be tackled. For example, the security of the application, the reliability of the smart contract and the double-spending problem lying in the heart of blockchain technology. While the first prototype of UI and library is implemented, testing has not yet been done.

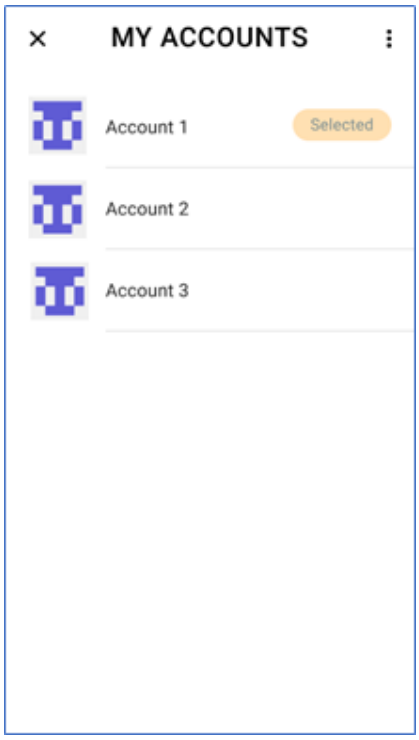
The project plan is designed carefully to make sure the project will be delivered by April 2021. More research needs to be done on maintaining the credibility of HKDT when problems like dishonesty persist as this is a problem bundled with the mechanism of blockchain.

Appendix A

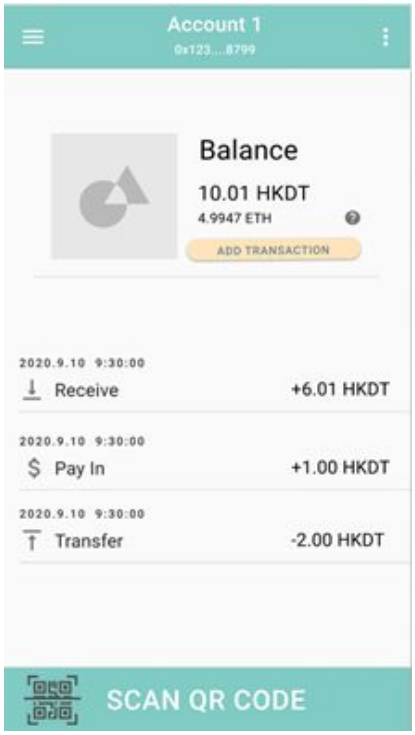
UI Design of the HKDT Wallet

Login Page:

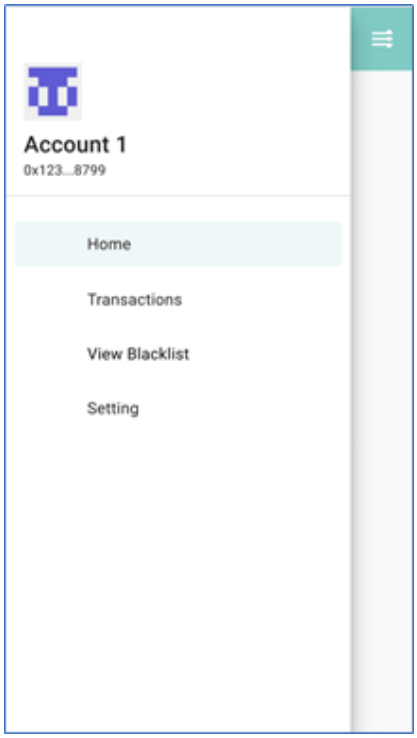
Account Switching Page:



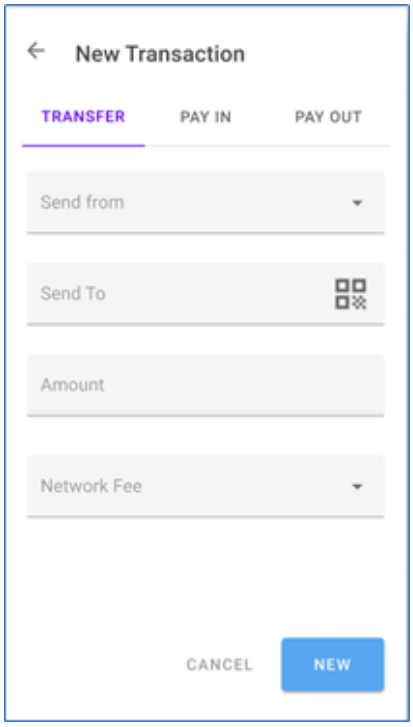
Home Page:



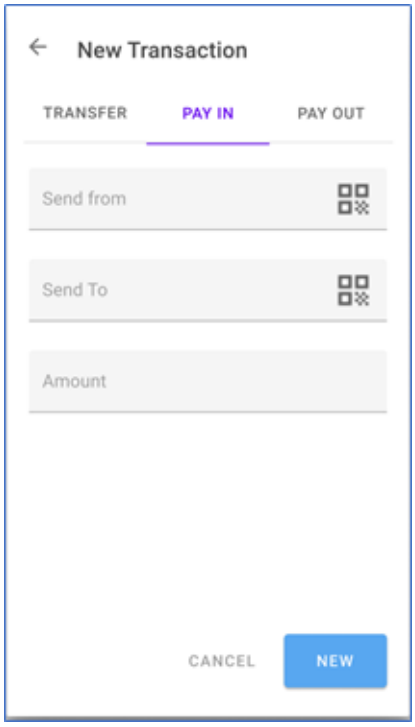
Side panel:



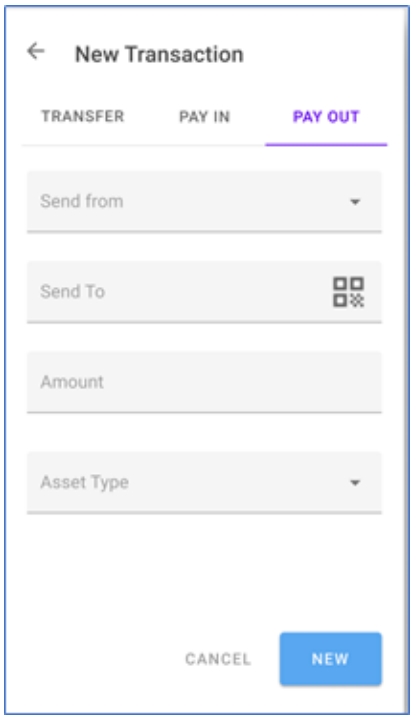
New transaction - Transfer:



New transaction – Pay In:



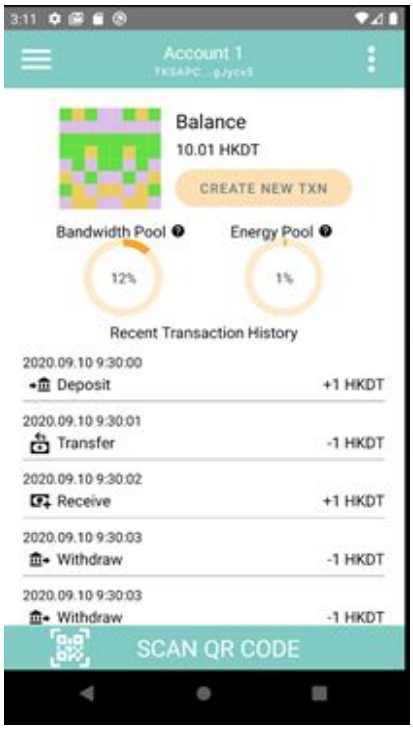
New transaction – Pay out:



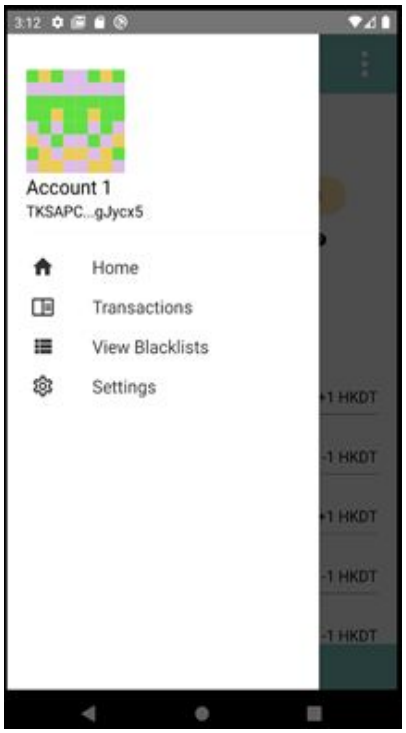
Appendix B

UI Implementation of the HKDT Wallet

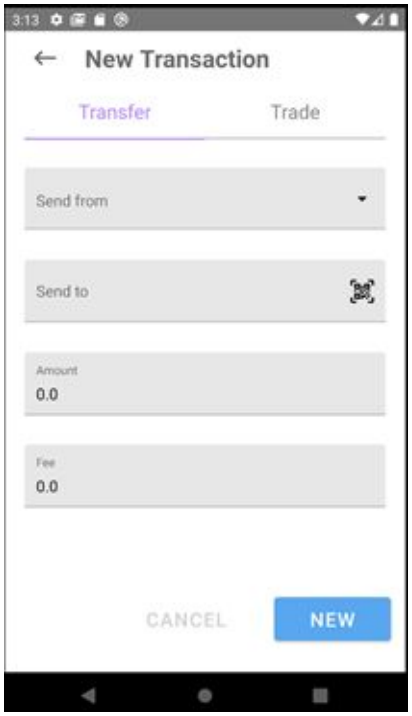
Home Page:



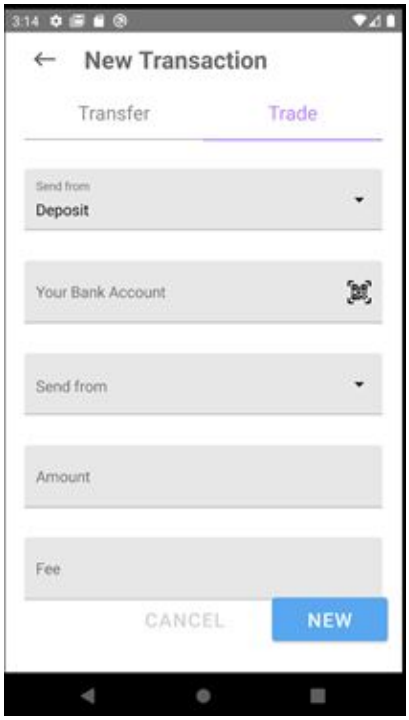
Side Panel:



New Transaction – Transfer:



New Transaction – Trade:



Appendix C

Implementation of Blacklist in USDT Smart Contract [26]

```
pragma solidity ^0.4.18;

import "./Ownable.sol";

contract BlackList is Ownable {

    /////// Getter to allow the same blacklist to be used also by other contracts (including upgraded
Tether) ///////
    function getBlackListStatus(address _maker) external constant returns (bool) {
        return isBlackListed[_maker];
    }

    mapping (address => bool) public isBlackListed;

    function addBlackList (address _evilUser) public onlyOwner {
        isBlackListed[_evilUser] = true;
        AddedBlackList(_evilUser);
    }

    function removeBlackList (address _clearedUser) public onlyOwner {
        isBlackListed[_clearedUser] = false;
        RemovedBlackList(_clearedUser);
    }

    event AddedBlackList(address indexed _user);

    event RemovedBlackList(address indexed _user);

}
```

```
pragma solidity ^0.4.18;


/**
 * @title Ownable
 * @dev The Ownable contract has an owner address, and provides basic authorization control
 * functions, this simplifies the implementation of "user permissions".
 */
contract Ownable {
  address public owner;


  event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);


  /**
   * @dev The Ownable constructor sets the original `owner` of the contract to the sender
   * account.
   */
  function Ownable() public {
    owner = msg.sender;
  }


  /**
   * @dev Throws if called by any account other than the owner.
   */
  modifier onlyOwner() {
    require(msg.sender == owner);
    _;
  }


  /**
   * @dev Allows the current owner to transfer control of the contract to a newOwner.
   * @param newOwner The address to transfer ownership to.
   */
  function transferOwnership(address newOwner) public onlyOwner {
    require(newOwner != address(0));
    OwnershipTransferred(owner, newOwner);
    owner = newOwner;
  }

}
```

References

[1] Nakamoto, Satoshi. Bitcoin: A peer-to-peer electronic cash system. Manubot, 2019.

[2] Cryptocurrency Market Capitalizations. (2020). Retrieved January 12, 2021, from https://coinmarketcap.com/

[3] Hileman, Garrick, and Michel Rauchs. "Global cryptocurrency benchmarking study." Cambridge Centre for Alternative Finance 33 (2017): 33-113.

[4] Chuen, David LEE Kuo, Li Guo, and Yu Wang. "Cryptocurrency: A new investment opportunity?." The Journal of Alternative Investments 20.3 (2017): 16-40.

[5] Ycharts.com. Bitcoin Average Transaction Fee. (n.d.). Retrieved October 15, 2020, from https://ycharts.com/indicators/bitcoin_average_transaction_fee

[6] Metz, C. (2017, June 03). The Grand Experiment Goes Live: Overstock.com Is Now Accepting Bitcoins. Retrieved October 15, 2020, from https://www.wired.com/2014/01/overstock-bitcoin-live/

[7] Smith, A. (2014, December 11). Microsoft begins accepting Bitcoin. Retrieved October 15, 2020, from https://money.cnn.com/2014/12/11/technology/microsoft-bitcoin/index.html

[8] Mita, Makiko, et al. "What is stablecoin?: A survey on price stabilization mechanisms for decentralized payment systems." 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI). IEEE, 2019.

[9] Team, Maker. "The dai stablecoin system." URl: https://makerdao. com/whitepaper/DaiDec17WP. pdf (2017).

[10] Moin, Amani, Emin Gün Sirer, and Kevin Sekniqi. "A Classification Framework for Stablecoin Designs." arXiv preprint arXiv:1910.10098 (2019).

[11] Christidis, Konstantinos, and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things." Ieee Access 4 (2016): 2292-2303.

[12] Wang, Shuai, et al. "An overview of smart contract: architecture, applications, and future trends." 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018.

[13] Mohanta, Bhabendu Kumar, Soumyashree S. Panda, and Debasish Jena. "An overview of smart contract and use cases in blockchain technology." 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2018.

[14] Haigh, Trevor, Frank Breitinger, and Ibrahim Baggili. "If i had a million cryptos: Cryptowallet application analysis and a trojan proof-of-concept." International Conference on Digital Forensics and Cyber Crime. Springer, Cham, 2018.

[15] Tether Limited. (2014). Tether (USDT). Retrieved October 19, 2020, from https://whitepaper.io/coin/tether

[16] Yu, H. (n.d.). International legal business solutions - Global Legal Insights. Retrieved October 19, 2020, from https://www.globallegalinsights.com/practice-areas/blockchain-laws-and-regulations/hong-kong

[17] React Native · A framework for building native apps using React. (n.d.). Retrieved October 20, 2020, from https://reactnative.dev/

[18] Danielsson, William. "React Native application development." Linköpings universitet, Swedia 10 (2016): 4.

[19] Hansson, Niclas, and Tomas Vidhall. "Effects on performance and usability for cross-platform application development using react native." (2016).

[20] TRON Foundation：Capture the future slipping away. (n.d.). Retrieved October 20, 2020, from https://tron.network/

[21] TRON Foundation. (2018, December 10). TRON: Advanced Decentralized Blockchain Platform. Retrieved October 20, 2020, from https://tron.network/static/doc/white_paper_v_2_0.pdf

[22] Ethereum Average Transaction Fee. (n.d.). Retrieved October 21, 2020, from https://ycharts.com/indicators/ethereum_average_transaction_fee

[23] The TRON FAQ¶. (n.d.). Retrieved October 21, 2020, from https://tronprotocol.github.io/documentation-en/faq/

[24] Huang, Z. (2020, April 19). Bitcoin: Hong Kong's First Approved Crypto Fund Targets $100M. Retrieved October 22, 2020, from https://www.bloomberg.com/news/articles/2020-04-19/hong-kong-s-first-approved-crypto-fund-is-targeting-100-million

[25] Cryptocurrency deposit processing times. (n.d.). Retrieved October 23, 2020, from https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times

[26] TRONSCAN: TRON BlockChain Explorer. (n.d.). Retrieved October 24, 2020, from https://tronscan.org/

[27] Resource Model. (n.d.). Tron. Retrieved October 24, 2020, from https://developers.tron.network/docs/resource-model

[28] M. Christina. (2020, December 23) "Top 5 Prominent Token Development Platforms in 2021," *DEV Community*, Retrieved January 22, 2021, from https://dev.to/merlinchristin2/top-5-prominent-token-development-platforms-in-2021-fpi.

[29] PwC (December 2019). In depth A look at current financial reporting issues. Retrieved January 22, 2021, from https://www.oecd.org/finance/The-Tokenisation-of-Assets-and-Potential-Implications-for-Financial-Markets.pdf