Loyalty Program using Blockchain

Osman Sönmeztürk
Dept. of Computer Engineering
Izmir Institute of Technology
Izmir, Turkey
Email: osmansonmezturk@gmail.com

Tolga Ayav
Dept. of Computer Engineering
Izmir Institute of Technology
Izmir, Turkey
Email: tolgaayav@iyte.edu.tr

Yusuf M. Erten
Dept. of Computer Engineering
Izmir Bakırçay University
Izmir, Turkey
Email: murat.erten@bakircay.edu.tr

***Abstract*—The traditional loyalty systems usually offer people benefits in a specific sector. The users usually need to stay within the loyalty system for a long time and accumulate points in order to win rewards which may not be very interesting for them most of the time. Additionally, users usually do not prefer to share their personal information to join these loyalty systems due to privacy concerns. It has, therefore, been observed that the number of customers in the loyalty systems is decreasing day by day. To reduce these drawbacks a loyalty program which complies with ERC20 standards was proposed in this study using tokens based on the Ethereum blockchain. Using this new generation loyalty system, users can convert their earned tokens to Ether in the market and they can receive services or products with the accumulated tokens according to their interests from any supplier that has been contracted by the manufacturer. Additionally, users in the designed system do not need to carry many different cards, it is adequate to have only one Ethereum wallet. Furthermore, users do not need to share any personal data to join the loyalty system. Suppliers can also request Ether from the manufacturer for the tokens they have accumulated from the members of the loyalty system. The proposed loyalty system has been implemented and presented in this study.**

***Keywords*-loyalty system; erc20; Ethereum;**

I. Introduction

Manufacturers and markets have been developing and implementing the loyalty system since the late 19th century to encourage customers to buy their products. Traditionally, there are 6 types of loyalty programs which are Punch Cards, Points, Tiered, Fee-Based, Cash Back, and Coalition loyalty programs. Today, with the increasing role technology plays in our lives, many studies are carried out on the feasibility and transfer of these systems to the blockchain.

According to a recent research, the number of members in the loyalty systems reached 3.8 billion in 2016. However, 57% of their members think that it takes a long time to win a reward in the loyalty programs, and 53% think that the awards given by these systems are not very interesting [1]. In another survey where 177 people participated, 86% found the idea of earning ether in a loyalty system based on blockchain technology attractive and 67% stated that they would prefer to keep their money as ether [2]. It is foreseen that the loyalty program based on blockchain and cryptocurrency will solve this important problem by instantly uploading awards to the user accounts.

Although the idea of blockchain was originally proposed as a payment system, it quickly became popular in other areas due to its difficult to break and easy to maintain structure. Blockchain technology is not only seen as a cryptocurrency but it has also been used in areas such as IoT, security, and voting systems.

Blockchain technology has also been used in loyalty systems and it is considered to be a good candidate to meet the new expectations of the users of the loyalty systems and make these systems more reliable. With the developments in the blockchain technology, the loyalty system allowed transactions not only with one manufacturer but several manufacturers, and different loyalty systems were established on the blockchain network to meet business to business needs.

In the blockchain system, there is a protocol between the sender and the receiver based on asymmetric cryptography. When the coin holder transfers funds to another user in the blockchain system, the hash of the transaction information is calculated using the hash function, and is encrypted with the sender's private key. Transaction information and encrypted hash output are sent to the recipient. Using the public key generated by the sender, the receiver can decrypt the encrypted hash output and also calculate the hash function. The hash output is then compared with the received hash value if the two hash outputs are equal, one can be sure that the transaction information is correct and has not been changed [5].

These transactions are then included into blocks to be added to the chain. The blocks are generated by the nodes called the miners and depending on the system there is a difficulty function involved which ensures that the block which contains the hash code of the previous block and cannot be tampered with. The difficulty function used in Ethereum is Proof of Work (PoW) like most of the other crypto currencies [6].

Most loyalty systems use tokens instead of cryptocurrencies because more resources such as time, money and energy are consumed to generate the cryptocurrencies usually due to a consensus mechanism employed and also the tokens are stored in an array on the network and act as a transaction ob-

978-0-7381-0495-9/20/$31.00 ©2020 IEEE
DOI 10.1109/Blockchain50366.2020.00074

ject on the chain. The cryptocurrencies cannot be produced and delivered to customers quickly according to the supply quantity because it takes a long time to obtain. In addition, the difficult-to-obtain crypto currencies cannot be cancelled or undone but the expiration date can be determined for a token or different ID's can be given to tokens based on their usage area. Using tokens, the manufacturer can define a specific marketing strategy. Producers can use these tokens to attract their customers into their loyalty systems, and the users may feel the intended win-win relationship better. The customers in the system do not have to carry a lot of cards with them physically, they need to have a single wallet which may be used for keeping records of the Ethers they owne as well as tokens which may be from different suppliers hence different contracts. The creation fee of the tokens is close to zero and there only exists a fee to run the function on the Ethereum chain. Another advantage is that the given tokens can be destroyed if the customer owns an excessive amount of token in a short time period or gain tokens through illegitimate ways. If the supply and demand increase sharply and it becomes necessary to generate tokens, the owner of the smart contract only needs to run one function for mint. Although the manufacturer is the one who decides and determines the market strategy, the high number of tokens in the market will decrease the value of the token. Hence there is a market-economy based control over the tokens produced by the manufacturer.

In our study, the rules used in the Ethereum network were followed to meet the security issues and other standards. Everything was implemented in the Ethereum network and users can run the smart contract functions within the loyalty system through the web interface as common to most systems.

As can be seen in the proceeding sections, there are many studies that have integrated blockchain into loyalty systems. Overall, the results from these studies show that the customers are satisfied with the new generation loyalty systems. Our motivation to develop the loyalty system proposed in this study is to ensure the satisfaction of the customers as well as the satisfaction of the manufacturers and the retailers in the system.

In this work, it is aimed to briefly introduce the research studies on the new generation loyalty systems. The loyalty tokens produced were derived from the ERC20 standards and were written using a smart contract with the stable Solidity version [3]. Ethereum chain was preferred for the system, due to its large developer community and fast block production. In addition, a proposal for the win-win relationship between the market, the customer and the manufacturer has been presented within the established loyalty system.

When designing a loyalty system, a smart contract is also required in the blockchain network. The smart contract must be generated to create a commitment between seller and buyer. Some of the high-level languages used to create smart contracts are Solidity, Vyper, and Bamboo. However, smart contracts created with these languages cannot be run directly on the Ethereum nodes. First, to enable communication between the contract and the node, the contract must be compiled and the low level opcode obtained this way is sent to the Ethereum Virtual Machine (EVM). Running the created opcode, EVM allows the smart contract to access the Ethereum nodes.

In the next section we shall attempt to explain some of these proposed solutions and proceed to explain our solution in the remainder of the text.

II. Related Work

There has been other studies carried out to apply blockchain technology to the loyalty systems. Some of the more relevant ones are summarized in the proceeding paragraphs.

The article in [7] investigates the effect of blockchain technology on loyalty systems. With the help of self-determination theory, the needs in the fields of economy, autonomy, competence, and relatedness are defined. The results are categorized into three dimensions: economic utility, psychological self-fulfillment, and social interaction. The name of the tokens used in the established theoretical loyalty system is 'Yun point' and these tokens are purchased from a platform that holds the mint authority called 'gege point'. Companies and customers can transact with the generated token and transfer among themselves but must pay the minter a commission to use the tokens.

In [8] a token-based peer review payment system has been proposed as an incentive for the reviewers. In accordance with the smart contract between the author and reviewer, the author gives 100 peer-review coins (prc) to the system and the system receives 1 prc for its sustainability and distributes 33 prc to 3 reviewer teams. The good evaluation of the reviewed article and its timely completion in the review processes can increase the earned token and the earned prc tokens can be converted into tradable money on the exchange market. The system is intended to encourage the reviewers to complete the process on time and with maximum care.

In [2], a loyalty strategy was defined for students and staff to do their shopping from the selected 12 retailers. While the user is using the system, he/she reads the barcode defined on his phone and then sends the pico user ID and timestamp information to the cloud. The information in the cloud reaches the Ethereum node through the developed platform. The loyalty system tested at the University of New South Wales (UNSW) works by converting the collected stamps into ether. If the customer has accumulated 10 stamps, the loyalty platform requests ether from the Ethereum chain and the received ether is loaded into the customer's account. In order to keep the participants longer in the loyalty system, marketing strategies have been introduced. There are promotions such as a double win campaign in the second

week and a $ 5 ether gift for the first registration in the system.

National Taiwan University of Science and Technology produced NTUSTokens on a private Ethereum chain in order to increase students' technology awareness and on-campus activities [9]. A private chain is preferred because the users log into the system with their student IDs and the chain only works inside the campus. Users can earn tokens by going to the library or doing sports, and can send these tokens to their friends. It is expected that this system will help the students adapt to campus life quickly . Moreover, paying attention to the use of electricity or consuming water resources consciously enables them to earn tokens. On the other hand, the negative behaviours of the users dissipate tokens and is reflected as a punishment.

Another loyalty system shows resemblance to the system created in our study, but the blockchain-based cross-organizational integrated platform has not adopted any standards, and the Solidity version is also outdated [10]. According to the article, goods providers and token providers can register to the system and value different products with certain amount of tokens and users can shop with their tokens as much as they can. The goods providers within the established system have a private chain among themselves. They can also access the database and users through the web server. The installed system functions like a blockchain-based shopping site. It was also mentioned that the established system is suitable for the operation of shopping malls and airline companies.

The loyalty system proposed in [11] was established on the neo chain and tokens are called Promotion Asset Exchange. In the proposed loyalty system, customers can earn tokens by reading the QR codes of the products they buy with their mobile phones. The mobile phone connects to the neo blockchain with neon.js and requests a token. The customer can use the collected tokens to buy products from the merchants and the merchant accumulates the tokens they receive in return for the product they give to the customers and gets money from the contracted manufacturer in exchange for the token. The system objectives mentioned are similar to those of the system we have proposed, but the used blockchain technology and the operation of the system are completely different.

III. The Proposed System

Our proposed system is based on creating a contract on the Ethereum network and producing tokens based on this contract. The manufacturer of the tokens, which is usually a company initiating the loyalty system, can create as many tokens as necessary, and can make the created TECH tokens (that is what we call our tokens) available to the exchange market or just keep them in the loyalty system. Just like cryptocurrencies, the produced tokens can be transferred between ethereum wallets. Customers in the system do not have to carry different cards for different loyalty systems of each company, an Ether address will suffice. The customer can earn the TECH Tokens produced by the manufacturer as they purchase products from the contracted suppliers. Moreover, depending on the strategy of the manufacturer, customers can buy the token from the exchange market or exchange it with cryptocurrencies. Users can purchase services or products by giving TECH Token to any supplier included within the system. A sample wallet is shown in Figure1 and Figure 2.

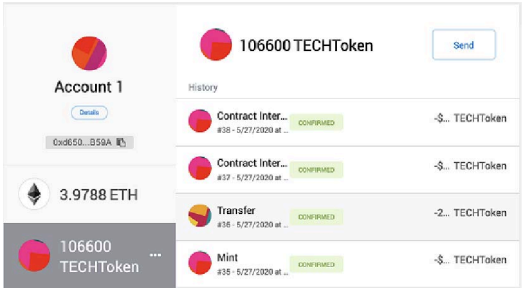

Figure 1. Transactions shown in the manufacturer wallet

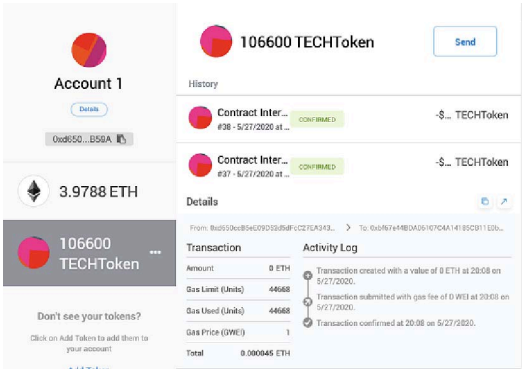

Figure 2. Transactions details shown in the wallet

The wallet belongs to a manufacturer which is also the minter of the tokens. As depicted in the figures the wallet contains both Ether and tokens. the tokens belong to a certain contract namely TECH tokens. Figure 2 show the details of one of the transactions.

The members of the loyalty systems also have their wallets and they can keep their Ether and TECH tokens in the same wallet as shown in Figure 3. From the figure it can ve observed that the user has transferred tokens three times, still has 1183 TECH Tokens in his/her wallet as well as 3.48 Ethers. If the user was a member of another loyalty system managed through a different contract then the token transactions from that contract would also be seen in this wallet.

The loyalty system built on blockchain is accessible and transparent to everyone, and thus the customers can keep track of their earned tokens. In addition, it is useful to

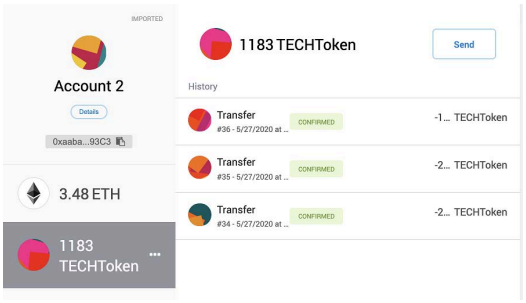

Figure 3. Transactions shown in the loyalty system member wallet

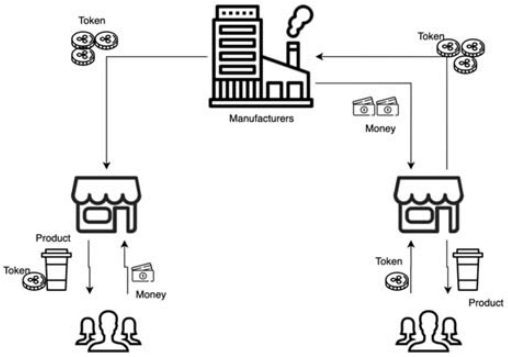

Figure 4. TECH token flow in the system

use blockchain not only for the customers but also for the manufacturers. Based on the manufacturer's market strategy, token production and earnings balance can be determined easily.

The TECH tokens are produced on the Ethereum chain following the standards which are derived from the ERC20 [12]. The Solidity programming language was used to write smart contracts. This language, which is formed by the blend of object-oriented C++ and JavaScript, is compiled in Ethereum virtual machine and run on the Ethereum chain. Each transaction written in Solidity has an operating fee on Ethereum. Therefore, smart contracts were created taking the amount of gas consumption into account and avoiding loops.

The ERC20 standards in the library have been followed and new features have been added to create the tokens.

At the initial stage, the manufacturer, who has a contract in the designed system, starts sending the tokens to the suppliers in the loyalty system. The customers are issued these tokens when they make purchases from suppliers affiliated to the system. Afterwards, the suppliers start receiving tokens instead of money from customers for their products and services, as customers start spending the tokens they have accumulated. In addition, customers can exchange tokens to Ether from the exchange market. Moreover, the accumulated tokens are given to the manufacturer and the manufacturer gives Ether to suppliers according to the accumulated token

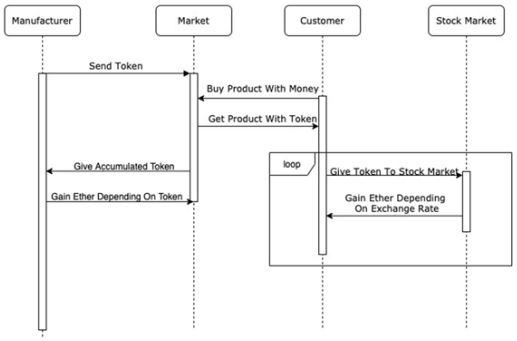

Figure 5. Time Diagram of the Token Flow

amount in the markets at a determined exchange rate which may also change according to the market rules. Thus, the contracted suppliers make profit in the system. The general structure of the loyalty system can be seen in Figure 4, and the flow diagram is depicted in Figure 5.

The more the customer spends the more value the tokens gain and as tokens gain value, the customer tends to spend more in the designed system. As a result, the customer continues to remain inside the loyalty system as a user. Through this, the manufacturer can produce more services and products and make agreements with more retailers. The suppliers within the system can sell more products and services as users shop and earn ethers back. Since customers earn as much as they spend, they are included in the system and are encouraged to spend more as they stay in the system.

All token transactions and transfers can be monitored via the Rinkeby network [13]. In fact, Table I and Table II are transactions extracted from this network. A web interface is necessary for the manufacturer to perform the operations.

It was not preferred to run the new generation loyalty system on the Ethereum main net, because the ethers traded on the main net are not as easy to produce as on the Rinkeby network and are produced with consensus algorithms by the miners. However, in order to produce ethers in the Rinkeby network, ethers are requested only with Faucet [14].

Because the amount customers spend is directly proportional to the amount they earn, not only the continuity of the customers within the system is provided, but also the manufacturer and the market make profit. As a result, the targeted win-win-win relationship in the new generation loyalty system is established.

IV. Implementation

Before starting to set up the system, alternative chains were studied to find out which would be more suitable for the desired system. Research has shown that the development of the Ethereum chain can be done more quickly since there are already many tools developed. In addition, the high number of transactions per second is an important factor in

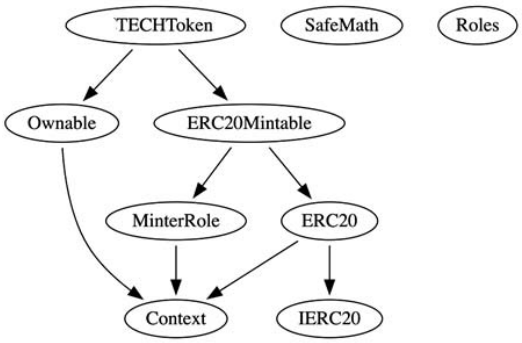

Figure 6. TECH Token components

the selection of Ethereum network [15]. In order to make transactions on the Ethereum chain, a smart contract was written to meet the requirements of the loyalty system.

The token types such as ERC20, ERC721, and ERC77 on the Ethereum chain are derived from the standard Open Zeppelin library. ERC20 token standard is used in the established system. The functions that have been added to the smart contract have determined the working methods and rules of the award system. In addition, since it requires gas to be used for each process, the loop functions were not used in the development and the problem was solved by mapping where the loop operation was required.

TECH Token consists of a combination of two libraries, five smart contracts and an interface that enables the production of tokens as can be seen in Figure 6.

Safe Math library prevents overflow errors during arithmetic operations and limits the use of gas. The Roles library assigns the role to the given addresses and has 3 basic mapping functions. It can only use the functions of the desired contract with 'using' keyword. The Context contract, which is the most commonly used smart contract, was used to prevent direct access to the owner information and data. The MintRole smart contract was produced by inheriting the context contract and the Role library was used. The MintRole contract is the contract where transaction authorizations are created on tokens, and IERC20 is the contract where the standards of tokens are determined and were used as an interface. In IERC20, there are functions that need to be overloaded in the parent contract. ERC20 contract adds functionality by overloading the functions specified in the interface. Also, token arithmetic operations were performed using the Safe Math library in the ERC20 contract. The ERC20Mintable contract is derived from the MinterRole and ERC20 contracts, and it is the contract in which the mint function is defined. The ownable contract contains functions that the owner of the contract can use. Using this, the owner of the contract can delete his or her own role or can transfer to other addresses. TECH Token is made of Ownable and ERC20Mintable contracts and Safe Math library is used for the arithmetic operations. Finally, with the combination of smart contracts and libraries created in a modular structure, usable TECH Tokens are produced within the system. Due to this modular structure, not only the code quality and security of the smart contract spent have been increased but also the gas consumption has been reduced further.

As a final step, the TECH Token contract combines the contracts created and includes modifications according to the desired market strategy. The name, symbol and decimal value of the token which is used to compile the smart contract for the first time was created. Then the address was mapped to the boolean and its value was assigned to the awarder's variable, while unsigned integer was mapped to the address and its value was assigned to the awards variable. At the end of these operations, there is the constructor method that mints the token for the owner of the smart contract. The giveAward function first checks whether the token to be given belongs to the owner of the contract and then checks if there is sufficient balance for the transaction. If it fulfills all the mentioned requirements, it performs transfer with superclass transfer function.

The revokeAward function was used to disable previously given tokens. In order to run this function, the contract must be owned. The isAwarder function checks whether the given address is authorized to give tokens. The onlyAwarder function is run as a modifier and checks the role of the given address. The addAwarder function allows dealers in the system to assign tokens. The deleteAwarder function disables the dealers' ability to issue tokens. The addAward function separates the token types by enumerating the generated tokens. PayBack function is the function that gives dealers ether based on the number of tokens collected by the dealers. This function takes the amount of token address of retail and exchange rate and sends the ether to the retail address. Since currency transfers on the Ethereum chain are carried out over the Wei coefficient, the ether sent during the calculation is multiplied by the Wei coefficient and divided by the exchange rate. The TransferMerchantToMarket function is used to send a large number of tokens to the markets, and the tokens produced by the merchant cannot be given to markets as a reward, so they are carried out by another transfer outside the award system.

A. Installation of the system

The installation of the system can be divided into 3 main phases. The first stage is to create and test the smart contract online compiler, then to migrate the written contract and run the web interface on the local chain, and finally run the written contract on the Ethereum Rinkeby chain. Contracts do not need to be rewritten to add new features to the created loyalty system, as the new feature can be transferred to contracts created as an inheritance. The online compiler, which is remix editor, was used to test the contract written for the first phase. Developments were made in

10 ether accounts prepared for the test. Using the online compiler, the smart contract was converted to bytecode with the stable version of Solidity and optimized to minimize gas consumption.

After the first stage, the smart contract was deployed to the local chain. In order to create the local chain, Ganache was used which generates fake blocks and puts the transaction information in the blocks, thus creates the chain in the desired port. A smart contract was deployed to the created fake blocks using truffle, that compiles the smart contract by looking at the config file and sends the generated bytecode to the Ethereum virtual machine.

The frontend written using React.js was connected to the blockchain running locally and therefore, the loyalty system has started working on the local chain. Web3.js has been integrated into the system in order to enable the frontend to communicate with the blockchain. At this stage, the development of the frontend was emphasized. The frontend was developed with the structure of the components. The Application Binary Interface, which is created after the smart contract is compiled, is integrated into the React.js and called with function names and inputs from web3 are given. The currentProvider function in web3.js was used to allow the inputs of the functions to access blockchain nodes. However, in the beta version of the web3.js, installing and using web3.js with the node package manager can be problematic.

After this stage, the system was deployed to the online Rinkeby chain, in which anonymous people from all over the world can control the flow of tokens across the network. An application programming interface (API) number has been created for deployment that can be connected to the Rinkeby network node by getting a free membership via Infura. The config file of the Truffle has been changed so that the contract can access the Rinkeby network. Also, the hdwallet provider npm package was installed and this package was used to access the user's ether account. Using the config file, Truffle first accessed the ether account and then deployed the contract to the Rinkeby network through the accessed account. In addition, Infura can be used not only for the Rinkeby network but also for Ethereum networks, that is, main net can be reached with the same config.

Table I below shows the ERC20 token flow on the smart contract. It is taken from Rinkeby network and can be seen by anyone. In this table, there is information about the transaction hash block number, Unix timestamp, DateTime, sender address, receiver address and quantity of tokens. Some transactions on the main contract, which is (0xbf67e44bda06107c4a14185cb11e0b0033b2d80f) can also be seen in this table. This is publicly accessible at the site given in reference [16].

All transactions of the selected addresses (0xaaba235fB01f054F596f62798efB762B899F93C3) can be seen in Table II. The table gives information on when the transactions were made, on which smart contracts it was made, the sender and receiver addresses, and the amount of tokens traded. For this reason, users in the new generation loyalty system can easily follow the awards won. Also accessible from reference [17].

Discussions and Conclusions

In this study, we have reinterpreted the traditional loyalty programs from a blockchain perspective. The proposed system involves a manufacturer of the tokens to be used to verify that a purchase has been made and an award has been earned, the retailers or service providers to issue these tokens to the end customer and the users of the loyalty system who are basically the customers. The system allows the manufacturer of goods to be the minter of the tokens, or the manufacturer may use an independent token manufacturer (minter).

The installed system works on Ethereum chain and smart contracts have been created with Solidity language. The relationship between Ethereum nodes and the user is provided by the frontend, which is written using React.js and connected to the Ethereum network with Web3.js. The local blockchain was created with Ganache and was used during the development phase and after the completion of the developments, the smart contract was migrated to the Rinkeby network.

The system was designed in order to provide advantages to the manufacturer, suppliers, and customers. Within the system, the manufacturer becomes the owner of the TECH Token contract and distributes the tokens to the suppliers which agree to sign a contract with them. When customers purchase products or services, suppliers give TECH Tokens to them along with products and services. This feature allows users to spend the rewards they earn as they wish. In this way, the user can use the accumulated tokens for their next purchase or can buy ether according to the current ether token exchange rate in the market. In terms of suppliers, they can also receive ether based on the number of tokens accumulated. Since customers in the system tend to shop more to earn more tokens, the suppliers sell more products and consequently manufacturers produce more products. As a result, it was aimed to make profit for the manufacturer, market, and customers in the established loyalty system and a tiny economy was established.

The loyalty system can be followed over the Rinkeby network and can be seen by everyone. However, since it is costly to migrate and run the smart contract, the loyalty system mentioned in this study is currently not available to the main net. The proposed loyalty system can be moved to the main net in the future if sponsored by companies. In order to increase the value of TECH Tokens in the exchange market, statistical studies can be conducted on the stock market token standards and token amounts. Due to the modular structure of the smart contract, new features can be

Table I
ERC20 TOKEN FLOW ON THE SMART CONTRACT

Txhash	Blockno	Unix Timestamp	Date Time	From	To	Quantity
0x9b02358d833b9902ba1a5409ca472d30dba3e9276b44a2f3f618ca2cb75f977b	6562841	1590598084	5/27/2020 16:48	0x0000000000000000000000000000000000000000	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	100,000
0xdc7459ae519da05cb5e1e3ebc488299725fa4598bc5c9b2c02f9ed142e295a0f	6562847	1590598174	5/27/2020 16:496246377	0x0000000000000000000000000000000000000000	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	10,000
0x8b701dbe2ca70ea40ec01580a00bf7bf8979cbbbe0fb0666a3ccdf463e573842	6562866	1590598459	5/27/2020 16:54	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	0x3cb2fac5351fe7dd5bb4b4e4ad8775cc2f312e2c	2,000
0x2ee34de50f87919394479f3f4f3622ecf671b3c79cd8931c4b82e2a4ab1e19fe	6562881	1590598684	5/27/2020 16:58	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	0xaaba235fb01f054f596f62798efb762b899f93c3	3,000
0xe3bf08df3074a3c4c74b4188e524f926a64426ae76f8f34b42f19f8d27a8f9eb	6562886	1590598759	5/27/2020 16:59	0x0000000000000000000000000000000000000000	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	2,300
0x6c17a0fb94eef46e5ee9f9b8b57ea6bfad9212d73951ce81f07d37fd2c6127c2	6562889	1590598804	5/27/2020 17:00	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	0xaaba235fb01f054f596f62798efb762b899f93c3	2,000
0x229637948dc279ceb7d7287e57044b38fdaaa3d6f3c83b1f5f1d8b1990116721c	6562900	1590598969	05/27/2020 17:02:49	0xaaba235fb01f054f596f62798efb762b899f93c3	0xea42941efc36c36ac56c2e9f7e78587b4455b093	2,300
0x049deec478c9e6cbf2293c002556559f2e76523d531d93cbc36092906786b04d	6562935	1590599494	05/27/2020 17:11:34	0xaaba235fb01f054f596f62798efb762b899f93c3	0x3cb2fac5351fe7dd5bb4b4e4ad8775cc2f312e2c	216
0xfdc356917db5620950888246104389c4a8dbfad3b821d6ca824894b93dd9727b	6562939	1590599554	05/27/2020 17:12:34	0xaaba235fb01f054f596f62798efb762b899f93c3	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	1,300

added in line with the requests received from companies and the structure can be extended.

REFERENCES

[1] The-cma.org. (2020). The 2017 COLLOQUY Loyalty Census. [online] Available at: https://www.the-cma.org/Contents/Item/Display/327325 [Accessed 15 Feb. 2020].

[2] Shelper, P., Lowe, A., Kanhere, S. S. (2018). Experiences from the Field: Unify Rewards-A Cryptocurrency Loyalty Program. In Proceedings of the Symposium on Foundations and Applications of blockchain, University of Southern California, Los Angeles, USA, 2-7.

[3] GitHub. (2019). OpenZeppelin Contracts. [online] Available at: https://github.com/ OpenZeppelin/openzeppelincontracts /tree/master /contracts [Accessed 13 Dec. 2019].

[4] Underwood, S. (2016). Blockchain Beyond Bitcoin. 59(11), 15-17.

[5] Merkle, R. C. (1989). A Certified Digital Signature. In Conference on the Theory and Application of Cryptology, 435, 218-238. Springer, New York, USA.

[6] Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S. (2016). On the Security and Performance of Proof of Work Blockchains. In Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 3-16.

[7] Wang, L., Luo, X. R., and Xue, B. (2018). Too Good to Be True? Understanding How Blockchain Revolutionizes Loyalty Programs. In Proceedings of the 24th Americas Conference on Information Systems, New Orleans, Louisiana, USA, 1-10.

[8] Avital, M. (2018). Peer review: Toward a Blockchain-Enabled Market-Based Ecosystem. Communications of the Association for Information Systems, 42(28), 646-653.

[9] Daz, M., Mass, J., Prez-Villegas, A., Sagals, A. (2017). Smart Token for the Campus Using Blockchain. National Taiwan University of Science and Technology, Taiwan.

[10] Liao, C. H., Teng, Y. W., Yuan, S. M. (2019). Blockchain-Based Cross-Organizational Integrated Platform for Issuing and Redeeming Reward Points. In Proceedings of the 10th International Symposium on Information and Communication Technology. Ha Noi Ha Long, Vietnam, 407-411.

[11] Bulbul, S., Ince, G. (2018). Blockchain-based Framework for Customer Loyalty Program. 3rd International Conference on

Table II
SAMPLE TOKEN TRANSACTIONS ON A WALLET

Txhash	Unix Timestamp	Date Time	From	To	Value	Contract Address	Token Name	Token Symbol
0x2ee34de50f87919394479f3f4f3622ecf671b3c79cd8931c4b82e2a4ab1e19fe	1590598684	5/27/2020 16:58	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	0xaaba235fb01f054f596f62798efb762b899f93c3	3,000	0xbf67e44bda06107c4a14185cb11e0b0033b2d80f	TECH Token	TECH Token
0x6c17a0fb94eef46e5ee9f9b8b57ea6bfad9212d73951ce81f07d37fd2c6127c2	1590598804	5/27/2020 17:00	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	0xaaba235fb01f054f596f62798efb762b899f93c3	2,000	0xbf67e44bda06107c4a14185cb11e0b0033b2d80f	TECH Token	TECH Token
0x229637948dc279ceb7d7287e57044b38fdaa3d6f3c83b1f5f1d8b1990116721c	1590598969	5/27/2020 17:02	0xaaba235fb01f054f596f62798efb762b899f93c3	0xea42941efc36c36ac56c2e9f7e78587b4455b093	2,300	0xbf67e44bda06107c4a14185cb11e0b0033b2d80f	TECH Token	TECH Token
0x049deec478c9e6cbf2293c002556559f2e76523d531d93cbc36092906786b04d	1590599494	5/27/2020 17:11	0xaaba235fb01f054f596f62798efb762b899f93c3	0x3cb2fac5351fe7dd5bb4b4e4ad8775cc2f312e2c	216	0xbf67e44bda06107c4a14185cb11e0b0033b2d80f	TECH Token	TECH Token
0xfdc356917db562095088824610 4389c4a8dbfad3b821d6ca824894b93dd9727b	1590599554	5/27/2020 17:12	0xaaba235fb01f054f596f62798efb762b899f93c3	0xd650ccb5ee09d52d5dfcc27ea3437551b64eb59a	1,300	0xbf67e44bda06107c4a14185cb11e0b0033b2d80f	TECH Token	TECH Token

Computer Science and Engineering, Sarajevo, Bosnia and Herzegovina, 342-346.

[12] Docs.openzeppelin.com. (2020). OpenZeppelin Documentation. [online] Available at: https://docs.openzeppelin.com/ openzeppelin/ [Accessed 14 Feb.2020].

[13] Rinkeby.etherscan.io. (2020). Rinkeby Testnet Explorer. [online] Available at: https://rinkeby.etherscan.io/ [Accessed 13 Feb. 2020].

[14] Faucet.rinkeby.io. (2020). Rinkeby: Authenticated Faucet. [online] Available at: https://faucet.rinkeby.io/ [Accessed 13 Feb. 2020].

[15] Lee, D. R., Jang, Y., Kim, H. (2019). Poster: A Proof-of-Stake (PoS) Blockchain Protocol using Fair and Dynamic Sharding Management. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 2553-2555.

[16] https://rinkeby.etherscan.io/token/0xbf67e44bda06107c4a14185cb11e0b0033b2d80f

[17] https://rinkeby.etherscan.io/address/0xaaba235fb01f054f596f62798efb762b899f93c3/#tokentxns