UxVote – Blockchain-Based E-Voting System for Secure Electronic Voting

BAK Vinsura[1], RMVD Bandara[1], ADAI Gunasekara[1], B Hettige[1] and GAI Uwanthika[1]

[1]*Department of Computer Engineering, Faculty of Computing, General Sir John Kotelawala Defence University, Ratmalana, Sri Lanka.*

vinsurakumuthu@gmail.com

Abstract *: Voting is a process of group decision-making or opinion-gathering that can be utilized to resolve any ideological disagreements. Voting on paper is still the most popular method. However, this traditional method of collecting votes is quite expensive and employs paper ballots. As a solution to this, a very secure and transparent solution is a necessity, which should also ensure the privacy of the participants. An e-voting system can be taken into consideration as a remedy to the problems that the traditional voting system currently has, and one of the technologies that are most suited for use in highly secure situations like this is blockchain. A hashing technique serves to strengthen the security of a blockchain, which is a decentralized system. Peer-to-Peer networks and a decentralized timestamping server make it difficult to manipulate or alter the data in this system. In this paper, we are presenting a safe voting system that was created using blockchain technology that allows voters to select one candidate from an existing group for major elections (e.g.: presidencies) and general elections. In this system, we used the Ethereum network, Ganache blockchain, and the Solidity programming language to create and test an example e-voting application as a smart contract for the Ethereum network. The records of ballots and votes will eventually be stored on the Ethereum blockchain. Voting requests are handled by the consensus of all Ethereum nodes and can be made by users straight from their Ethereum wallets. This agreement offers an open environment for electronic voting. With the help of this system, voting may be done more securely and affordably online.*

Keywords: *Blockchain, Smart Contracts, E-voting, DApps Ethereum*

1. Introduction

With the proliferation of bitcoin worldwide, blockchain technology has become a considerable part of people's lives. Although the use of blockchain in its early years was only for monetary transactions and trade, studies show that it can be used in many areas beyond that when considering the high transparency of blockchain technology into consideration. Furthermore, in this P2P-based system, there is no requirement for a central authority to authorize or complete the operations. Consequently, all types of structural information are retained in this distributed chain; including monetary transactions. The system is kept secure with the aid of specific cryptographic techniques. A large amount of information such as property records, marriage certificates, patient medical records, etc. can be captured using this approach/system after making any necessary alterations (Wood, 2014). A few years after Bitcoin's establishment, another cryptocurrency, the Ethereum coin (Ether), was developed. Now, Ether distinguishes the blockchain in its real sense by showing that this blockchain technology can produce software that can store data using the aforementioned structure. The blockchain contains immutable code that is used to enforce smart contracts (described below). Once they are written, they can neither be (illegally) erased nor altered. As a result, they are capable of functioning properly, independently, and transparently perpetually, without any outside influences (Maaten, 2004).

Voting is a method of group decision-making or opinion-gathering. Other ideological problems can also be resolved using this technique. This voting procedure is used to choose candidates for various positions in every Sri Lankan election. Presently, the paper-based system is the most popular and conventional means to cast a ballot.

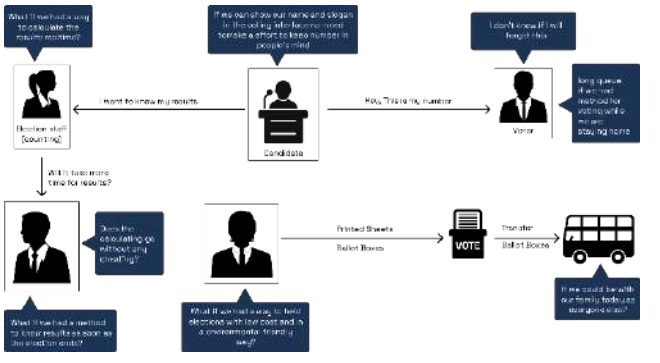

Figure 1 : Rich picture, how the current system process works and concerns at every stage

This time-worn technique has a relatively high cost because votes are collected using paper ballots. The ballot, the ballot box, the counting procedure, security, transportation, and many other expenses must be accounted for. Currently, this counting procedure is conducted manually. It takes time to carefully process and count all of these voter pamphlets at counting stations, and occasional incidents of election violence have also been heard. Cases of fraudulent voting and the violations that result from them directly undermine election transparency and public trust in democracy.

Consequently, there is a need for a reliable and secure voting mechanism today.

Comparing digital voting (or electronic voting) to conventional paper-based voting methods, digital voting is a better cost-effective alternative. However, governments are hesitant towards these approaches because of security concerns and people's lower technological awareness. However, a safe electronic voting system maintains transparency and security to the necessary level and reduces these expenditures, while also boosting the legitimacy of election outcomes (Tarasov and Tewari, 2017).

Nevertheless, blockchain technology may handle numerous difficulties outside digital trade, as a result of its distinctive distributed and secure concept. It might be a good option for projects involving electronic voting. E-voting is the subject of in-depth research, and numerous systems have been tried out and even employed for a while. However, only a few implementations are trustworthy enough to be used today. There are many successful examples of online surveys and polls, but we cannot make the same statement about online elections for businesses and governments. This is mainly because democratic administrations, which are the most popular form of government in the modern world as well as in Sri Lanka, depend heavily on free and fair elections. Additionally, in democratic nations like Sri Lanka, a robust election system that promotes openness is highly appreciated. A decisive election system that offers privacy and transparency is also what democratic societies prefer the most. Due to these circumstances, we are suggesting our blockchain based e-voting system, ‘UxVote’.

Section II of this research paper focuses on the global need for a reliable and transparent e-voting system that addresses the flaws and limitations of the traditional voting system. Section III describes the complete technology behind a blockchain-based voting system, built using the Ethereum blockchain. Section IV elaborates on the web application that is used to host this voting system, its functionalities and how this application would function in a real-life scenario step-by-step. The conclusive inferences made through the findings of this research are discussed in Section V.

2. Literature Review and Related Works

Our primary driving force behind this project is demonstrating the viability of a trustworthy e-voting system using blockchain while offering a safe voting environment and increasing the citizens' trust in the elections, while being an economically viable solution for our nation. As anyone with a computer or mobile phone can use e-voting, every administrative decision can be voted upon by the people. At the very least, the public's view will be more visible and available to managers and legislators. Most importantly, this will eventually lead to actual direct democracy for everyone. It's crucial because elections can be readily influenced or corrupted, particularly in small villages and even larger cities located in corrupt nations.

Additionally, large-scale traditional elections are exceedingly expensive over the long run, mainly if there are millions of voters and hundreds of geographically dispersed voting centers. Plus, voters may be out of their relevant residential towns, making it hard for that specific voter to participate in the election and potentially lowering turnout. If done correctly, electronic voting will be able to address these issues effectively (Lin and Espinoza, 2007).

E-voting is a much older notion than blockchain. But all of the examples that are now known used centralized computing and storage models. Estonia is one of the best countries which were able to use a comprehensive and entirely online voting system. E-voting was first discussed in the nation in 2001, and it was formally implemented by national authorities in the summer of 2003 (Braun, 2004). Their system can keep going due to several enhancements and changes to the initial plan. It is currently quite robust and reliable as reported because for person-based authentication, they make use of card readers and smart digital ID cards for citizens that are distributed by the government. There is a unique web portal as well as a related desktop application that citizens can use to participate in the elections by selecting their choice from the listed candidates and casting their votes, so that anyone with a computer, an Internet connection, and their ID card can cast a ballot online with ease (Lohrmann, 2020). However, the Estonian model has numerous shortcomings, despite its considerable success and recent elections' penetration rate of 30% (approx.). By its very nature, the centralized approach offers a single point of failure and is vulnerable to hacking and hijacking attempts. For instance, Distributed Denial of Service (DDoS) assaults might damage the servers, databases, or applications that are being used. If they are unable to change the data, the administrators of such a system may act maliciously and manipulate some important information during an election. Plus, with regard to scalability, even though it works well in a country with a small population like Estonia, it is difficult to guarantee that it will work in a country with a large population (e.g., China). And this ongoing requirement for an ID card and ID card reader will not work in a cost-efficient system either (Koç, Yavuz, Çabuk and Dalkılıç, 2018).

Switzerland is also one of the few nations partaking in the trend of computerized voting. Every citizen of Switzerland, a country renowned for its extensive democracy, who has reached the age of 18 is eligible to participate actively or inactively in elections that may be held on a wide range of issues and for a wide range of choices (Ladner, Felder and Schädel, 2008).

A thorough study article recommends a robust methodology for a blockchain-based electronic voting system. The use of an intermediary unit between the voters’ wallets and the administrators’ wallets as well as the use of two different coin types for these intermediary coin (vote) transfers were additional countermeasures for voter privacy and vote anonymity that were explored by the designers.

Here, the intermediary unit gathers the coins (votes) given by the voters and converts them into another currency using that currency's wallet. Then, the intermediary unit delivers the new coins to their intended recipients (candidates). It is a very informative source, but it neither provides much information about the practical issues of implementation beyond the usage of Bitcoin and Zerocoin as the currencies, nor offers a thorough discussion of it.

Table 1: Feature comparison with existing systems

Features	Estonia	Switzerland	UxVote
User-Friendly UI	✓	✓	✓
Usage of Smart Cards	✓	✗	✗
Usage of Blockchains	✗	✓	✓
End to End Encryption	✓	✓	✓
2 separate channels for voting and user authentication	✗	✗	✓
Fully Automatic	✗	✗	✗

Our main objective is to concentrate on implementational tasks and develop our solution on a smaller scale in order to bring a general election process online, from elections for presidents, to elections for an organization's executive committee, and even for trivial matters such as student councils. The election process will be entirely online, so that everyone may participate in voting quickly in general elections (elections that allow voting for a single candidate at a time). We would like to conduct this in a way that everyone can monitor and keep track of the election process. Integrating online elections with the Ethereum blockchain technology is our main contribution to the idea of online elections. Only a few scholarly publications have examined the Ethereum blockchain as an e-voting option as of this writing (Meeser, 2017). The authors have presented a thorough and ostensibly safe protocol using the Ethereum blockchain, but their protocol involves complicated mathematical operations, and hence needs a lot of computing power, making it unsuitable for the Internet of Things (IoT). We created Ethereum smart contracts that enable vote verification and vote counting. Additionally, any Ethereum account can be added to the elections. The hash values of the accounts prevent identifying individuals.

3. Methodology

Following diagram illustrates the high-level system architecture of the system.

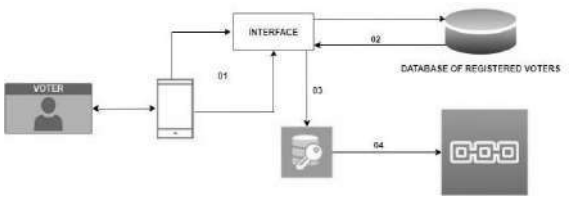

Figure 2 : Schematic diagram of the high-level system architecture

Request to vote (01): The administrator will approve the eligible voters with an Ethereum account number after the registration along with other valid information, such as their NIC, name and Ethereum account number. The system will allocate and grant permissions to the registered voters.

Casting a vote (02): Ethereum will create a token with an initial Boolean to cast the votes; however, the voters won't be able to do so until the token's value reaches 1. Voters will be given access to a simple and adaptable user interface for voting.

Encryption (03): To identify and validate system users, the system will produce a hash of the voter's name, confirmation number, and prior vote. This will guarantee the security and uniqueness of the information in the U-Vote e-voting system. The SHA one-way hash function is used to encrypt the data kept in the system. Data saved in the electronic voting system cannot be altered because they are linked to one another, ensuring the system's high level of security.

Adding the vote to the blockchain (04): Blocks are generated with the details required to store in the blockchain and blocks are linked to each other to produce the blockchain.

In this research, the Ethereum environment has been selected as the blockchain network and development platform of choice. Because of the strength of smart contracts, the Ethereum network offers a broader range of use cases than Bitcoin, which is exclusively designed to validate coinage transactions. These smart contracts allow many apps that typically run on a web server to be run without one. As a result, altering or damaging the intended software's source codes is challenging, if not impossible (Dzulfikar and Susanto, 2020).

All actions on the Ethereum network take place in real-time, or at least they are supposed to, and all blocks are added to the main chain in exchange for ethers, the network's cryptocurrency. These are awarded to the miners who carry out these time- and resource-intensive writing and validation processes. As indicated briefly above, we defined our smart contracts. These contracts were created using the Solidity programming language, which combines JavaScript and C++. The Ethereum network's peers execute smart contracts once every 15 seconds, and for them to be triggered, at least two additional users must validate them.

a) Why We Chose Blockchain Technology

We must find solutions to the following issues before we can conduct entirely online elections. The voting platform must be transparent, authenticated, and provable. We must ensure that the voters are actual people who use the legitimate identification documents we know for existing in electronic contexts. We must be able to show this at any moment, and we must ensure that our elections are as transparent as possible. As a result, we must collect and verify signed and timestamped election data. This ensures that once votes are cast, they shouldn't be able to be

changed. Additionally, we require individualism in elections to prevent voting for third parties.

These issues are not a concern when it comes to blockchain P2P technology. The blockchain allows us to define the necessary self-executing smart contracts, which is similar to creating code; including the definition of rules, objects and data models, so that contracts can begin to be carried out. Once established, smart contracts cannot be removed from the blockchain. A centralized organization is not required in the Ethereum network to deliver proof-of-work. The results of the contracts can be calculated by all of the peers without any influence. The Ethereum network has self-tallying capabilities (McCorry, Shahandashti and Hao, 2017).

In any case, using the original Ethereum network to try out new smart contract creation tools is expensive (because it costs some amount of ether to do so) and unnecessarily consumes a lot of memory. As a result, private Ethereum networks can be built and made accessible to developers so that they can test their programs without clogging up the main network. One such network is the Ganache personal blockchain, which we have also used in our research. Ganache comes with 10 free user accounts, which have 100 ETH (replica) for each account, which can be used for testing the smart contracts.

It should be emphasized that such test networks may have additional implicit or explicit rules or constraints.

To utilize a test network, users must first download a legitimate Ethereum wallet (the wallet which is used here is MetaMask) and then adjust the settings to change the connected network to the desired test network. MetaMask is available as both a browser extension and a mobile app. As this app is a web application, the browser extension was used here.

b) How We Programmed the System

In the Solidity programming language, the "Voter" object is specified as a struct. Voter was defined, and Voters were collected in an array. Voters have several characteristics, and they may have many more based on the use case situations. The variable named "isVoted" is a flag that indicates whether the voter has voted or not. Similarly, a variable called "vote" maintains the voter's preference among all candidates (defined as the candidate struct). “voterAddress” is a variable which is used to store the address of the voter account in the Ethereum network that is associated with an Ethereum wallet address.

The wallet address of the responsible person, who is the main administrator of the voting process, is stored in the address variable which is declared as "admin"; although he/she cannot interfere with ongoing (or completed) voting, he/she has the authority to initiate the voting process and the Voter objects that will be given to actual voters. According to the system design, the very first Ethereum account gets the administrator eventually when deploying the election.

The individual who has the Ethereum address approved by the admin has the power to vote under this contract.

A function called vote() exist in the system, which is accessible by any voter whenever they want to vote (until the deadline). Voters simply send the ID of the candidate on which they want to vote as a parameter, and their votes are logged as a result. This function initially determines who is currently attempting to perform the contract's function. Furthermore, if the individual has the right to vote and has cast his/her vote, the person is listed as having already voted, and the vote count of the candidate of his/her choice is increased by one.

An arrow function called getWinner() returns the winning candidate's ID. It does not complete the voting process, but it always returns the winning candidate when it is executed. This function examines each candidate, counts the votes, and then returns the winner of the entire voting process as of the time of the function call, but it does not end the election.

The blockchain entries (blocks) pertaining to the vote creation and casting procedures are shown in detail in Fig. 3 and 4. Anyone monitoring the network can access this data in the open.

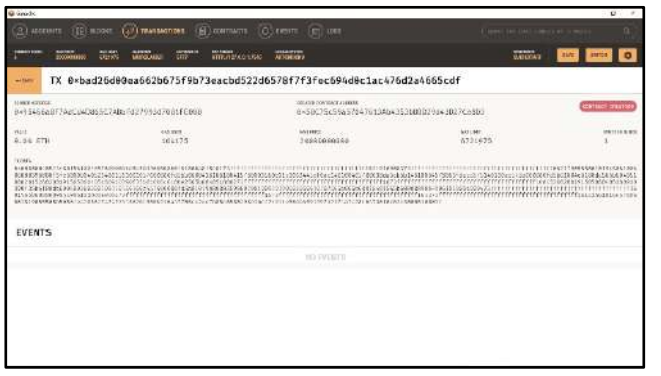

Figure 3 : The blockchain entries (blocks) pertaining to the vote creation

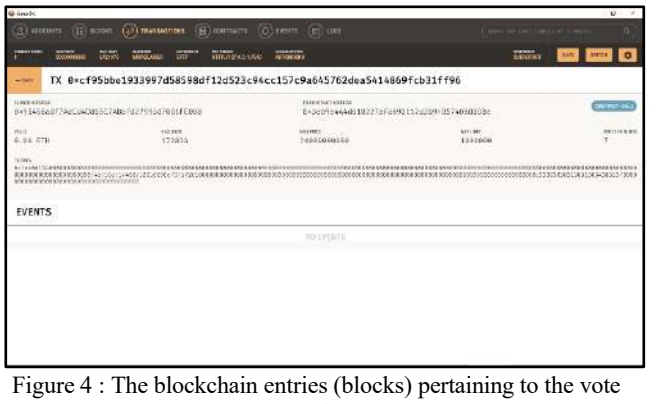

Figure 4 : The blockchain entries (blocks) pertaining to the vote casting

Fig. 5 and 6 depict, respectively, the reception of a cast vote and a request for the end election. On the other hand, only the person who have performed the action have access to the data shown in the screenshots in Fig. 5 and 6. This is because the person who have performed the action's wallet account is the only place from which these records may be directly acquired.

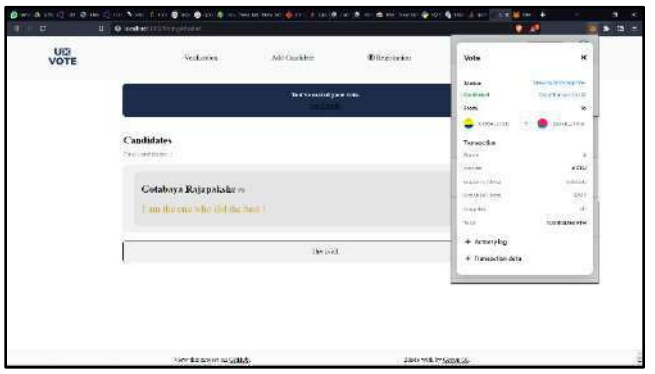

Figure 5 : Reception of a cast vote

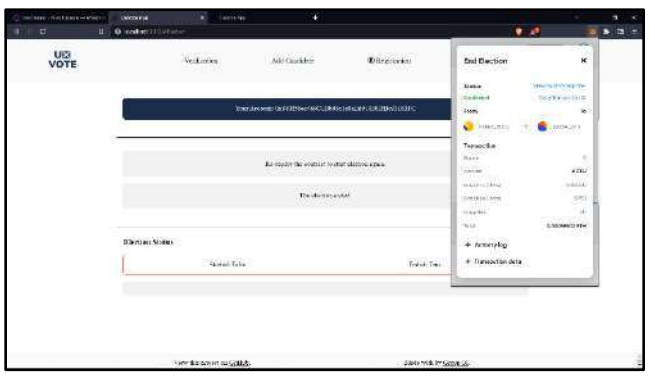

Figure 6 : Reception of an End Election Request

This project's scope is restricted to local presidency and general elections and polls. Millions of voters could present distinct challenges during a larger election. Presently, we cannot recommend using these contracts for nationwide elections because the Ethereum network's scalability is yet unknown and requires further study. These contracts are executed on the Ethereum blockchain, so this voting application may be utilized anywhere an Ethereum wallet can be run. The Ethereum wallet is currently supported on the Linux, OS X, and Windows operating systems. A voter should also have a small quantity of ether on hand in order to run the voting application and cast a ballot.

4. How The System Works

The very first Ethereum account receives the administrative functionalities after the election is deployed, per the system design. The owner of that account's private key will be granted administrative access, which will allow them to organize an election and add candidates to the database.

Only the administrator is able to see the interface below (Fig. 7 and 8), and from there, they can use it to set up an election. After entering the election data, such as the election title (05) and the organization name (06), the administrator is able to start the election process (08). The "Start Election" button can now be clicked to begin the election.

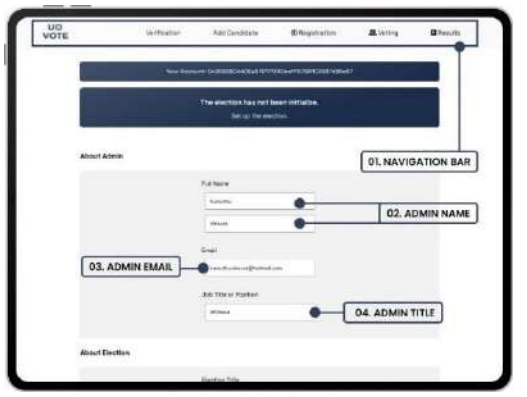

Figure 7: Election initialization (Page 1)

The button will not be available until you've finished steps 05, 06, and 07. When initiating an election, the administrator must supply their details, such as name (02), Email address (03), job title or position (04) from their end as well. Additionally, as an election cannot be started without the appropriate candidates added, a reminder to add them is displayed. The "Add Candidate" page will be displayed after clicking on "Add Candidate".

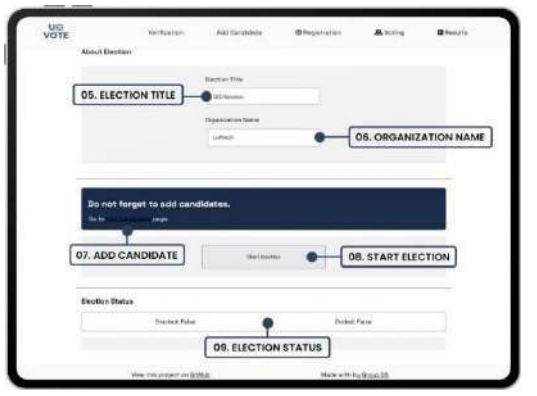

Figure 8: Election initialization (Page 2)

Without adding candidates who have been nominated to the election, admins cannot begin the election. As you can see in the interface above, there is a reminder mentioning the above case (Fig. 8). By using the navigation bar to visit the "Add Candidate" page or by using the link below the reminder, the administrator can add candidates both during and before the election setup. Both methods of adding candidates will take the administrator to the same interface, as displayed below (Fig. 9).

According to the system design, candidate adding functionalities are also available to the admin only, and candidates can be added to the election by completing the steps 10, 11, and 12. When initiating an election, the administrator must supply candiates' details, such as name (10), slogan/party (11). By selecting the "Add" option, the candidate can be included in the election. After steps 10 and 11, and only before the election begins, the "Add" button will be available.

The candidates who were added to the election will be listed below with their name, slogan, and index (13). This page also shows the total number of candidates who have been added to the election.

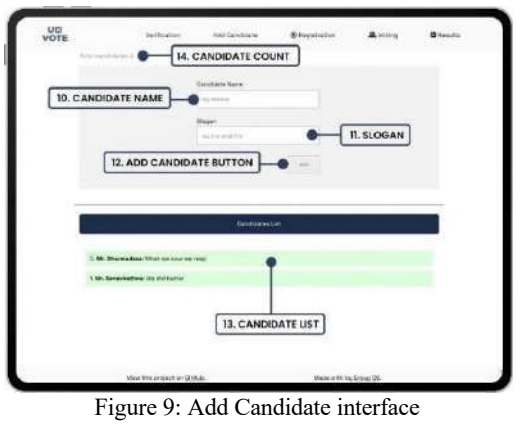

Figure 9: Add Candidate interface

In an emergency, if a user attempts to access the “Add Candidate” page by guessing the URL, an error message claiming that only the administrator has access to it appears. Because of this, users are unable to add candidates.

After completing the above steps, the admin gets the access to start the election, after which users get access to register as a voter for the election.

A card containing the voter's information will be sent to the administrator after a successful registration via their interface. The administrator can use their catalog to check the voter's information, and then they can simply authorize the voter for voting by pressing the “Approve” button, which is located on the same card. Administrators can still access the voter's information from their interface after giving them permission to vote, but they are not given the option to revoke that permission.

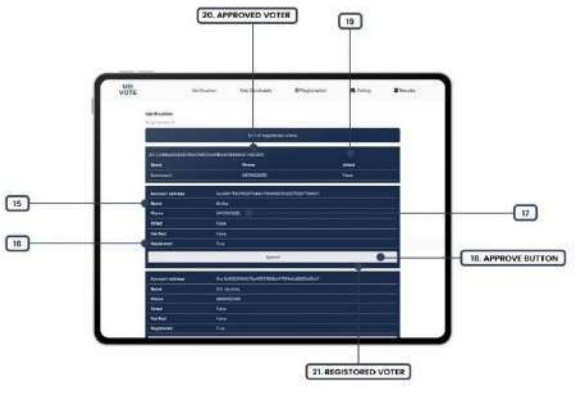

Figure 10: List of approved users, followed by the list of users to be approved

Here, the user's name (15), NIC number or the mobile number (17) that was recorded during registration will be displayed. Additionally, the administrator can also check the voter's status there, including whether they voted in their state or not. The interface shown in Fig. 10 will demonstrate the afore-mentioned functionalities. By clicking on the "Approve" button, the administrator can accept the qualified users after checking the users’ eligibility with their ledger manually. After selecting the “Approve” button, the “Verified” status displays “true” and levels up the user in the list as a verified voter.

After the election is set up, its details (22) are displayed as seen in Fig. 11 below. However, only the administrator can see the “End” button (24), demonstrating that only the administrator can end the election. Additionally, account details of users currently logged in (25) are also displayed here.

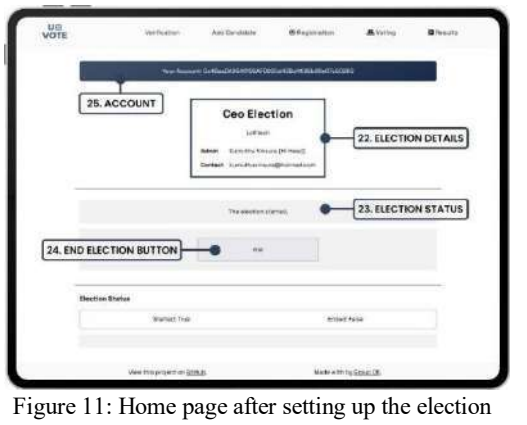

Figure 11: Home page after setting up the election

The user’s web application begins to display the user interface as Fig. 12 after the election has been launched by the administrator. After the election is launched, the system becomes open for the user to sign up for the election. Steps 26, 27, and 30 of the registration process for "UxVote" must be completed by the user by adding their details including the ETH account’s address (autofill) (30), name (26) and the mobile or NIC number (type of the number depends on the type of the election) (27). If the data entered is correct and valid, the “Register” (28) button will be accessible to the user. After the registration, the “Register” button will be visible again as the “Update” button (changed from “Register”) and before the approval of the administrator, the users can change the details they have entered, if they wish to do so. After the registration, the users can see their registered details at the bottom of the Registration screen.

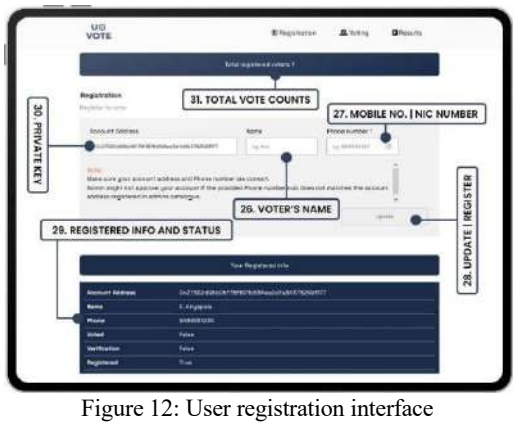

Figure 12: User registration interface

After the approval of the administrator, users can start voting by casting a vote for their preferred candidate. The

voting page (Fig. 13) consists of a list of candidates with their details. Every candidate has their name, candidate ID (33) and slogan/party (depends on the type of the election) (35) depicted on the list. Voters can cast votes for their respective candidate/s by selecting the respective “Vote” button (36). Additionally, the number of candidates who participate in the election (32) is also displayed in this screen as a usability improvement.

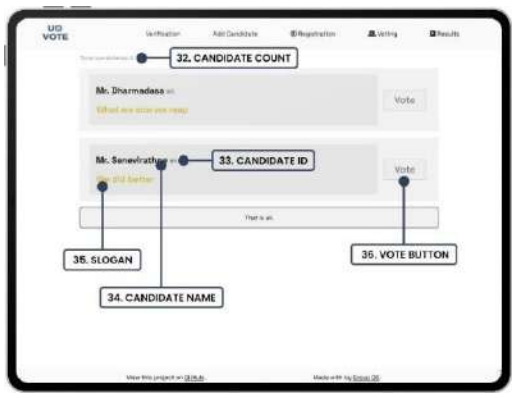

Figure 13: Voting page interface

A confirmation message will appear after selecting the "Vote" button to make sure that the vote is going to the right candidate, as shown in Fig. 14. By clicking on “OK” (37), the user can cast the vote to the respective candidate, and by clicking on “Cancel” (38) they can cancel their vote and select another candidate to cast their vote. After clicking on the “OK” button, the user cannot change their selection again.

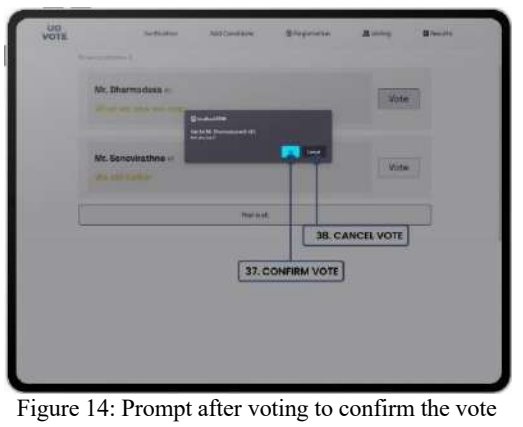

Figure 14: Prompt after voting to confirm the vote

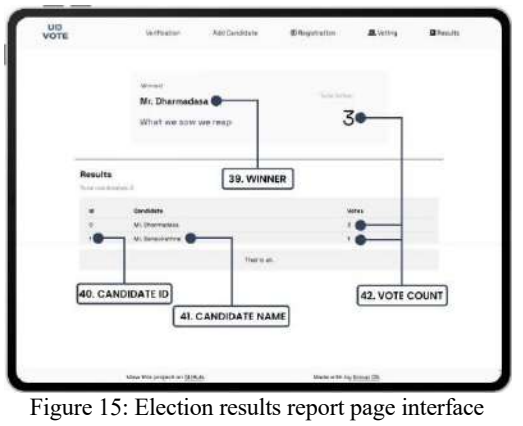

Figure 15: Election results report page interface

Fig. 15 above depicts the “Results” page, where the final results of the election will be displayed. The election's winner (39), who received most of the votes, is shown at the top of the results page, followed by a detailed breakdown of the election's overall results including the name (41), Candidate ID (40) and total number of votes (42) received by each candidate. Everyone who has the access to the result page can download a soft copy of the result sheet.

The notice “You’re not registered. Please register first” will appear with a link to the registration page if the user hasn't registered. The message "Please wait for admin to verify" will be displayed if the user registers but is not yet accepted by the administrator (Fig. 16).


Figure 16: Voting page interface before and after the user registration

This system comes with a few pages which include some notices for the users that indicates the current state of the election, as a usability improvement. The Voting page is visible to users only with their account number. Before the election has started, a message is displayed stating “The election has not been initialized. Please wait…”.

After the election is concluded by the administrator, the final results will be shown on this page as Fig. 15. The page will show a notification that reads, "The election is being conducted at the moment. Results will be displayed once the election has ended. Go ahead and cast your vote (if not already)”. If the user hasn't already voted, he or she can do so by clicking "Voting Page," which is presented as a link at the end of the message, which takes the user to the Voting page.

5. Conclusion

The “right to vote” is a fundamental human right, and as such, voting is a critical concern of people. Nowadays, since everything is done through electronic devices, voting may also be done in this manner, saving nations and organizations a fortune. This computerized technology improves the voting process's efficiency and security. By utilizing the power of the Ethereum network and the blockchain structure, we were able to successfully migrate the existing manual voting system to the e-voting system, and from it, to a blockchain platform while also addressing some of the basic difficulties that existing e-voting systems face. As a result of this project, the notion of blockchain and the security approaches it employs, such as immutable hash chains, can be adapted to polls and elections. This accomplishment could pave the way for other blockchain

applications that affect every element of human nature. At this point, Ethereum and smart contracts, which made one of the most revolutionary breakthroughs since the blockchain itself, helped to transform blockchain from a limited perception of a cryptocurrency into a broader solution-base for many Internet-related issues of the modern world and may enable the global use of blockchain structure and its associated technologies.

E-voting is still a contentious issue in both political and scientific circles. Despite the existence of a few extremely effective examples, the majority of which are still in use, many more attempts were either unsuccessful in providing the security and privacy elements of a regular election or had substantial usability and scalability concerns (Hao and Ryan, no date). On the contrary, blockchain-based e-voting solutions, such as the one we developed using smart contracts and the Ethereum network, address many security concerns, such as voter privacy, integrity, verification, non-repudiation of votes and transparency. However, there are some aspects that cannot be addressed purely through the blockchain, such as voter authentication (on the human level, not the account level), which requires the integration of additional mechanisms, such as the use of biometric factors like fingerprints, face recognition etc.

The importance of decentralized systems is evident when considering the risk that storing registrations in a centralized location possess. This can always give officials the opportunity to physically view the voting records, which could lead to corruption and cheating from the authorities' end. Furthermore, in today's linked world, with the concept of the Internet of Things (IoT), many non-computer digital devices are expected to gain Internet access. It is important to note that, aside from phones, computers and tablets, many other common objects; even cars, are also now able to access the internet. Consequently, building a huge, distributed network which can reserve the required processing power will not be a problem in today's world. Additionally, such a system may not be suitable for important or official elections. The Diffie-Hellman procedure, which also presumes the use of random numbers and public/private key pairs, reportedly enables the holding of a "two-round" referendum with some ballot privacy.

The author would like to apply UxVote to all of Sri Lanka's election processes as the next phase in the research. Furthermore, this should integrate and include a user authentication module which makes use of biometrics in authenticating users.

References

Braun, N., Chancellery, S. and West, B., 2004. Electronic Voting in Europe: Technology, Law, Politics and Society. *E-Voting: Switzerland's projects and their legal framework–In a European context*. Bonn: Gesellschaft für Informatik, pp.43-52.

Hao, F. and Ryan, P., 2017. *Real-world electronic voting*. 1st ed. CRC Press, pp.143-170.

Lin, G. and Espinoza, N., 2007. *Electronic Voting - Arguments in Favor*. [online] Stanford Computer Science. Available at: <https://cs.stanford.edu/people/eroberts/cs201/projects/2006-07/electronic-voting/index_files/page0001.html> [Accessed 22 August 2022].

Maaten, E., 2004. Towards remote e-voting: Estonian case. *Electronic Voting in Europe-Technology, Law, Politics and Society*, 47, pp.83-100.

Lohrmann, D., 2020. *Could Estonia Be the Model for Secure Online Voting?*. [online] Government Technology. Available at: <https://www.govtech.com/blogs/lohrmann-on-cybersecurity/could-estonia-be-the-model-for-secure-online-voting.html>.

Tarasov, P. and Tewari, H., 2017. *THE FUTURE OF E-VOTING*. IADIS International Journal on Computer Science and Information Systems, 12(2), p.149.

Meeser F. L., "*Decentralized, Transparent, Trustless Voting on the Ethereum Blockchain*," 2017.

Koç, A., Yavuz, E., Çabuk, U. and Dalkılıç, G., 2018. *Towards Secure E-Voting Using Ethereum Blockchain*. In: International Symposium on Digital Forensic and Security (ISDFS). p.2.

Ladner, A., Felder, G. and Schädel, L., 2008. From e-voting to smart-voting - e-Tools in and for elections and direct democracy in Switzerland. In: Direct Democracy in and around Europe: Integration, Innovation, Illusion, and Ideology.

Dzulfikar, F. and Susanto, A., 2020. Implementation of Smart Contracts Ethereum Blockchain in Web-Based Electronic Voting (e-voting). Jurnal Transformatika, 18(1), pp.56 - 62.

McCorry, P., Shahandashti, S. and Hao, F., 2017. A smart contract for boardroom voting with maximum voter privacy. *International Conference on Financial Cryptography and Data Security. Springer, Cham*, pp.357-375.

Wood, G., 2014. Ethereum: a secure decentralized generalized transaction ledger. *Ethereum Project Yellow Paper*, 151, pp.1-32.

Acknowledgement

I would like to express my special thanks of gratitude to my Research Methodology lecturer Dr. B Hettige, Supervisor Mrs. GAI Uwanthika as well as Dr. ADAI Gunasekara (Dean, Faculty of Computing) who gave me the guidance and the support needed to conduct this wonderful project, through which I acquired much knowledge in various subjects such as cryptography.

I am also tremendously thankful for my parents and friends, especially Mr. MDPP Goonathilake, Mr. EMUH Ekanayake, Ms. GMT Amarasinghe and Mr. RM Aratchige who aided me in finishing this project within the limited timeframe.

Author Biography

[1] KUMUTHU VINSURA, is a pride of Richmond College & Computer Engineering undergraduate of General Sir John Kotelawala Defence University. Currently, he's employed as a Freelance Web Developer. He is the Webmaster of Member Activities Sub Committee of IEEE Sri Lanka Section.

His current research interests include blockchains, social tactics in Search Engine Optimization and psychology.

[2] VIDURA BANDARA, is an alumnus of Ananda College & Computer Science undergraduate of General Sir John Kotelawala Defence University. He's a university colorsman in several sports. Currently, he's employed at "The AI Team" as the Project and Operation Manager. He is the Section Student Representative of IEEE Sri Lanka Section.

His current research interests include natural language processing, neural networks, artificial intelligence and psychology.