AN EMPIRICAL ANALYSIS OF USING BLOCKCHAIN TECHNOLOGY IN E-VOTING SYSTEMS

A Thesis Proposal
Presented to the Faculty of the
Department of Computer, Information Sciences and Mathematics
University of San Carlos

In Partial Fulfillment
of the Requirements for the Degree
BACHELOR OF SCIENCE IN COMPUTER SCIENCE

By
JAYSON V. CADIZ
NICOLE AMBER M. MARISCAL

ANGIE M. CENIZA-CANILLO, PhD
Faculty Adviser

December 2021

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

This is to certify that the authors are responsible for the work submitted in this capstone. The intellectual content of this capstone is a product of original work. Any assistance that the authors received in the preparation and work of the thesis itself has been acknowledged. In addition, the authors certify that the materials and literatures taken from other sources are properly quoted.

JAYSON V. CADIZ  December 12, 2021

NICOLE AMBER M. MARISCAL  December 12, 2021

APPROVAL SHEET

This capstone entitled, "**AN EMPIRICAL ANALYSIS OF USING BLOCKCHAIN TECHNOLOGY IN E-VOTING SYSTEMS**" prepared and submitted by **JAYSON V. CADIZ AND NICOLE AMBER M. MARISCAL**. in partial fulfillment for the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE**, has been examined and is recommended for acceptance and approval for ORAL EXAMINATION.

THESIS COMMITTEE

ANGIE CENIZA-CANILLO, PhD
Adviser

CHRISTIAN V. MADERZAO, M.Eng
Committee Chair

EAN JASON C. VELAYO
Member

KENNETTE J. CANALES
Member

PANEL OF EXAMINERS

Approved by the Committee on Oral Examination with a grade of **PASSED.**

CHRISTIAN V. MADERZAO, M.Eng
Committee Chair

EAN JASON C. VELAYO
Member

KENNETTE J. CANALES
Member

Accepted and approved in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE.**

CHRISTIAN V. MADERAZO, M.Eng.
Chair, Department of Computer, Information Sciences and Mathematics

December 12, 2021
Date of Oral Examination

ACKNOWLEDGEMENTS

We would like to express our gratitude to the Department of Computer and Information Sciences and Mathematics for allowing us to conduct this research and for guiding us in the revision of this paper, especially to our thesis adviser, Miss Angie Ceniza-Canillo, for her continuous support in the completion and presentation of this thesis paper. Thank you, dear teachers, for challenging us to explore and widen our knowledge on key concepts for our research.

We would also like to thank people who have been with us throughout the duration of this research. To our parents, we thank you for your unconditional encouragement and financial support for the conference registration. To our friends and classmates, we thank you for the moral support that boosted our spirits in continuing this research.

Lastly, we thank God Almighty for blessing us with perseverance and success through all the challenges we have faced in conducting this research. He has answered our doubts and prayers, leading to the completed output we have worked on and the person we have become now. To God be all the glory!

ABSTRACT

Having a secure voting system is mandatory for a democratic society. With the advancements of technology, the modern world shifted from using paper-based voting to electronic voting or e-voting with the hopes of increased convenience and better overall performance. Unfortunately, an e-voting system is still vulnerable to security issues such as external and internal risks especially due to its centralized nature. A potential solution to this is the integration of blockchain technology, one of the emerging technologies of today that is defined as a distributed database technology wherein there is no central unit that takes complete control of the blockchain. Given its decentralized nature, blockchain is a promising and revolutionary technology that reduces cybersecurity risks and provides transparency. Hence, the researchers conducted this study with the aim to provide an empirical analysis of using blockchain technology in e-voting systems by specifically analyzing its performance and security in comparison to a centralized e-voting system. The researchers will develop both a Blockchain-based E-voting System (BEVS) and a Centralized E-Voting System (CEVS), and conduct automated testing on both systems to analyze their differences in performance and level security. The results show that the BEVS processes requests slightly slower than the CEVS, where a local private blockchain as the database of the e-voting system handles requests faster than a public blockchain network. However, the BEVS is more reliable when it comes to the efficiency of an e-voting system as it has a complete 0.00% error rate. The speed at which the BEVS may also be affected by heavy load, however, all requests were fulfilled, and no system crashes occurred. The BEVS also proves to be a more secure e-voting system with less vulnerabilities detected. This study will be beneficial for institutions that are in need of a more secure voting system. It will also be a significant contribution to the growing field of blockchain technology.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

The first chapter of this study presents an overview of the challenges faced with different election systems, along with a potential solution to these problems. This chapter also features the institutions that can benefit from the results of this study, as well as the limitations of this study.

1.1 Rationale of the Study

Voting is a key component of a democracy. Traditionally, a paper ballot system is faced with challenges, such as the inevitable possibility of human errors that affect the credibility of an election. To eliminate that, elections in many developing countries and organizations have shifted to electronic voting or e-voting. E-voting provides ease by a quick voting process and an automatic mechanism for producing results while eliminating the risk of human error in tallying votes. However, there are still security risks that revolve around e-voting systems. Most e-voting systems are implemented with a centralized architecture, making it susceptible to cyberattacks on the server.

In the Philippines, e-voting has been implemented since the 2010 national elections. COMELEC utilized automated election systems, wherein registered voters cast their vote by shading the ovals on printed ballots and feeding their completed ballots into vote counting machines (VCMs). Unfortunately, the implementation of these machines suffered from technical, procedural and security-related difficulties. In fact, the Philippines suffered its worst-ever government data breach barely a month before the 2016 elections, leaking personal information of over 70 million registered voters (Chi, 2016). This attack was claimed to have been made by the Anonymous Philippines in their efforts to highlight the vulnerabilities of the systems. The 2019 midterm elections also suffered from technical difficulties brought about by VCMs, ranging from paper jams to defective machines, that further delayed the voting process and even resulted in the manual counting and verification of ballots (Espina, 2019). As the

nation continues to suffer technical issues during the elections, educational institutions implementing online classes are now also faced with the challenges of e-voting.

In April 2021, the University of San Carlos – Commission on Elections (USC COMELEC) conducted its first ever online elections for the university's annual Supreme Student Council elections. According to the USC COMELEC Investigation Committee (2021), they received reports of technical difficulties experienced by students that led to their inability to vote. These included reports of student voters not receiving their one-time password (OTP) that were needed to vote, crashing of the voting site, failure of the site to produce ballot receipts and fluctuations of votes noticed by poll watchers (Webster, 2021). Due to these technical issues, the USC COMELEC declared the elections a failure. A special election was conducted shortly after a month. This shows how vulnerable a centralized e-voting system is, where the main server is susceptible to cyberattacks.

A potential solution to the difficulties of e-voting systems is the integration of blockchain technology. According to Yli-Huumo, Ko, Choi, Park & Smolander (2016), blockchain technology is a decentralized, distributed database technology wherein there is no central unit that takes complete control of the blockchain. It consists of blocks of data that are chained together, with each block having a cryptographic hash, wherein the current block should contain the hash value of its preceding block. Given its decentralized and anonymous nature, it is nearly impossible to tamper with data that have been added in the blockchain. A blockchain-based e-voting system significantly increases the level of security due to its validation process using the consensus algorithm, an algorithm that verifies the contents of a newly created block before adding it to the blockchain.

The objectives of this paper is to implement an e-voting system using blockchain technology and provide an empirical analysis of the blockchain-based e-voting system in comparison to a centralized e-voting system. Both a Blockchain-based E-Voting System (BEVS) and a Centralized E-Voting System

(CEVS) will be developed, and analyzed by conducting tests that simulate student elections in a university in order to evaluate and compare the performance and the security of the systems.

1.2 Statement of the Problem

1.2.1 General Objective

This study aims to provide an empirical analysis of using blockchain technology for e-voting systems based on the performance and security.

1.2.2 Specific Objectives

The research aims for the following:

1. Gather information and evaluate an e-voting system using blockchain technology.
2. Design and develop two e-voting systems, the Blockchain-based E-Voting System (BEVS) and the Centralized E-Voting System (CEVS).
3. Test and evaluate the functionalities of the systems.
4. Analyze the performance of the systems based on their performance and security.

1.3 Significance of the Study

The study will be of great importance and contribution to the following:

Government. The study will benefit the government as this study provides a solution to the pressing challenges with the current electoral system used with a centralized architecture.

Educational and Organizational Institutions. The study will benefit the mentioned institutions as this study provides an electoral system that can be implemented for conducting elections within the respective institutions.

Voters. The study will benefit voters participating in various kinds of elections, as the proposed e-voting system would be able to minimize the risks of vote fraud and protect the voter's identity. With the numerous efforts put into the implementation of e-voting systems, this study also provides a controlled voting process with the use of blockchain technology that would also contribute to the convenience and ease of the voting process.

Field of Blockchain. The study will be an addition to the potential applications of blockchain technology, that it is not only limited to applications involving cryptocurrencies.

Researchers. The development of the said systems will benefit the researchers as this study will allow them to explore the field of blockchain and develop and enhance practical skills in designing and building innovative systems.

Future Researchers. The study will benefit future researchers as it will serve as a guide and reference for related research works and further developments within the field of study, particularly in the field of Computer Science, E-Governance and Blockchain Technology.

1.4 Scope and Limitation

The study will implement a BEVS and a CEVS. The researchers will be evaluating the performance and security of the systems by conducting automated testing. The gathered results will then be used to compare both systems.

The study is limited to only the evaluation of the performance and security of the BEVS and the CEVS. The system performance will be evaluated based on specific performance metrics through automated testing where both systems will be tested with two database environments - a local and cloud environment. The security of the systems will be evaluated by discovering the systems' vulnerabilities and potential threats. This will be achieved by conducting a vulnerability assessment using vulnerability assessment tools and penetration

testing tools. Hence, both systems will be deployed in a local environment, and the testing will be done by the researchers.

In terms of development, the limitation of the proposed e-voting systems is that these will only be available for access in web browsers, specifically those that support browser extensions for the BEVS in particular, since it will use the MetaMask browser extension for the digital wallet to be used in casting of votes.

Both systems will have three modules, the Login Module, the Voting Module and the Results Module. In both systems, the Voting Module will have the same interface, but differ in the backend implementation. The Login Module in the BEVS will request for the input of a unique private key as account credentials, while the Login Module in the CEVS will request for the input of a voter's registered ID as account credentials.

CHAPTER 2
REVIEW OF RELATED LITERATURE

This chapter contains discussions on information that provide significant insight for the study. This chapter also contains previously done research works and their methodologies. This also presents the notable gaps that have yet to be researched, as well as note-worthy arguments against the study.

Voting Systems

The history of voting systems begins with paper ballots, also known as a secret ballot or Australian ballot, as it was used in Victoria and South Australia in 1856 (Eulua, Gibbins & Webb, 2020). It is a voting method where voters are handed printed ballots generated by election committees wherein they can anonymously mark their candidate choice. After this has been verified, the voters will then place their ballots inside a container called ballot boxes. Ballot boxes would remain unopened until the official period for counting of votes. Its major advantage is that it protects the voter and ensures their freedom of expression wherein only the voter knows of their electoral choice (Kuca, 2019). This system was eventually adopted across the world as it was simple in terms of literacy and acceptable in terms of the privacy and anonymity of the voters. With a paper ballot system, problems such as vote-buying, electoral fraud and dishonest vote counting still persist.

In the early nineteenth century in England, there was an increasing number of debates regarding electoral reform, including concerns about paper ballots (Krimmer, 2012). This led to a bill proposal by George Grote which included a mechanical voting machine to conduct casting of votes in a convenient manner. Voters using the machine would mark the ballots by punching holes into them. Although the proposal was rejected by the parliament, this became a basis for further developments. One of these developments was by a Polish inventor, Jan Jozef Baranowski, a machine that supported counting of votes based on the idea of the adding machines in Paris. The purpose of the

machine was to minimize human error during counting of vote results. Another invention was a machine by Morris Williams and Steuben Bacon that serialized paper ballots to eliminate electoral fraud. Eventually, the use of mechanical voting machines proved to have successfully reduced the amount of administrative work in comparison to traditional paper ballot systems.

A new form of voting, called electronic voting or e-voting, made use of electricity. In this method, ballot casting, recording and tabulation are done with the aid of computers. There were quite a number of proposals of e-voting machines in the early nineteenth century, but all of which, although seemingly interesting, were disapproved by the government as they believed that the use of e-voting devices would have changed the existing electoral procedures. Only later in the twentieth century did e-voting gain its recognition from the government. In 1974, a direct-recording electronic machine (DRE) was invented in the United States and it allows voters to vote by pushing a button next to the candidate of their choice. Eventually, DREs came equipped with tactile devices to aid disabled citizens to independently cast their vote (Abu-Shanab, Knight & Refai, 2010). The development of such spread across Europe, South America and Asia Nowadays, voting machines have even incorporated a touchscreen system. Along with the advancement of technology, here comes a new and more complex form of e-voting, Internet voting or i-voting. Security experts, however, worry that the Internet is not secure as denial-of-service attacks and even malware penetration could go undetected and compromise the electoral process. Of all nations developing i-voting projects, only Estonia maintained an i-voting option for its voters.

The main motivation behind e-voting is to promote secret voting, further limiting spoiled ballots brought about by human error. Advanced technology may also assist handicapped and vision-impared voters, as well as voters abroad who still wish to participate in elections. In contrast, there are also some challenges to e-voting. A centralized system could be rigged by dishonest administrators. Politicians may also fear losing vote shares due to the changes in the electoral process. In the eyes of the general public, it may also be difficult to fully embrace

these technologies that inherently do not disclose the actual voting process. Hence, voters cannot validate the vote themselves. Implementation of e-voting systems requires extra planning and knowledge provided a certain setting.

Centralized Systems

Centralized systems are systems that use client-server architecture with the central server as the single authority that retains total control over all data and permissions in the network (Seal, 2020). The primary advantage of this network architecture is efficiency, since all key functions operate and are managed in a central location. However, this system also has its limitations. Primarily, the availability of the network depends on the central server (Touron, 2019). Hence, there is a single point of failure. If the central server crashes, all connected nodes would be disconnected from the network and would not be able to access the data. In addition to these limitations, there is also a higher security risk. A centralized system is vulnerable to data manipulation (Atlam and Wills, 2019). Data may contain sensitive information about its users, and since that data is stored in one location, it is an easy target for a variety of cyberattacks. Data privacy is also an issue where service providers might sell information about their customers to marketing companies in order to analyze a node's behavior.

Blockchain Technology

Blockchain is a decentralized, distributed database technology for transaction and data management that aims to create an environment where no third party is in control of transactions and data (Yli-Huumo, et.al, 2016). It consists of blocks wherein each block in the chain consists of data, particularly transactions, and if a new transaction happens on the network, a record of that transaction is added to the ledger of each user. Distributed Ledger Technology (DLT) is a decentralized network that is run by multiple participants. Blockchain is a form of distributed ledger technology in which transactions are registered using a hash, which is an immutable cryptographic signature. This assumes that if a

single block in a chain is modified, it will be clearly clear that the chain has been tampered with. It is a method of storing data in such a manner that it is difficult or impossible to alter, hack, or trick the system. A blockchain system also relies on a consensus algorithm in order to ensure an agreement among its nodes. A consensus algorithm is a core component that dictates how a blockchain behaves, and it also determines its overall performance (Ferdous, Chowdhury, Hoque and Colman, 2020). A research by Mingxiao, Xiaofend, Zhe, Xiangwei, Qijun (2017) also states that different consensus algorithms play a crucial role in the safety and efficiency and stability of blockchain as it solves the problem of mutual trust amongst nodes in the blockchain network.

There are several blockchain types based on specific requirements. Firstly, based on accessibility, blockchains can be either of a public, private or consortium (Lin and Lao, 2017). Public blockchains allow anyone to read and write transactions and are fully decentralized. Most cryptocurrencies are implemented on public blockchains. On the other hand, private blockchains limit their access to individuals within a specific organization, usually for the purpose of auditing and records management. Consortium blockchains are also considered community blockchain wherein multiple organizations are allowed to read and write transactions onto the blockchain. Private and consortium blockchains are less decentralized and focus on a smaller group of users, making it more sustainable and cost-efficient than public blockchains (Endemann, Wladawsky-Berger, LaPointe and Yen, 2020).

Another categorization for its types is based on the need of authorization to participate in the blockchain which comes in three, permissionless, permissioned and hybrid blockchain. In a permissionless blockchain, everyone is allowed to join the blockchain network, while in a permissioned blockchain, only authorized participants are allowed to interact with the blockchain. In a hybrid blockchain, a node could be participating in both a permissionless and permissioned blockchain to facilitate inter-blockchain communication (Shrivas and Yeboah, 2018).

Blockchains can also be categorized based on its functionality and smart contract support. In a stateless blockchain, the system only focuses on transaction optimization and the chain functionality of verification of transactions by computing the hashes. A stateful blockchain, on the other hand, provides smart contract and transaction computing capabilities. Unlike the stateless blockchain that stands independently from smart contract logic, stateful blockchains support business logic.

Related Work

There are a number of studies proposed that present different methodologies in the implementation of blockchain based e-voting systems. These studies investigated the nature of blockchain technology to resolve the usual e-voting limitations, such as vulnerability to DDos attacks that weaken security and auditability of a voting system. A specific study created a Blockchain-based Online Voting System (BOVS) that made use of smart contracts run on a Ethereum blockchain powered by Ganache and MetaMask (Yi and Das, 2019). Similarly, another study also utilized smart contracts powered in an Ethereum blockchain for the electoral process. Their study also investigated different blockchain frameworks, such as Exonum, Quorum and Geth, that are suitable for implementing blockchain-based e-voting systems to achieve liquid democracy (Hjálmarsson, Hreiðarsson, Hamdaqa and Hjálmtýsson, 2018).

In a blockchain e-voting system, voters cast their vote by their unique private key. One implementation of a blockchain e-voting system made use of Ethereum as the voter's token. This token may contain any of the Boolean values, one (1), to signify their ability to vote, or zero (0), which occurs after voters have cast their vote. This specific study also allowed for a blank vote, which shows the voter's protest or dissatisfaction with the candidates (Patil, Rathi and Tribhuwan, 2018).

Other systems also made use of different blockchain technology related tools. A study was aimed at the elimination of forgery of votes as well as voter authentication, anonymity and non-repudiation with the use of elliptic curve

cryptography (ECC) in their proposed system (Yi, 2019). The hash values of each block were computed based on the SHA-256 algorithm. Similarly, a different study on the performance and scalability constraints of a blockchain e-voting system conducted experimentations that were implemented in three scenarios, the first two utilized a permissionless blockchain where participants could mine the vote into the blockchain, and the third one implemented a permissioned blockchain with designated nodes for miners and validators, simulating a public voting model (Khan, Arshad and Khan, 2019).

In Crypto-voting, a blockchain based e-voting system, the system made use of sidechain technology, a technology that extends a blockchain for additional features. It limits writing on the main blockchain to reduce the risk of entering false data, and it eliminates the need to add a new type of currency. This study makes use of a permissioned blockchain technology along with tools such as smart contracts and Shamir's Secret Sharing algorithm, to solve privacy vulnerabilities (Fusco, Lunesu, Pani & Pinna, 2018).

A different study designed a blockchain voting system that implemented points-weighted voting. Based on smart contracts and a private Ethereum chain, this study also made use of a Feedback Mechanism and Wilson Score to evaluate their system. Its results showed that there was an improved accuracy of voting results and the feedback mechanism greatly suppressed malicious voting behavior (Ma, Zhou, Yang & Liu, 2020) .

Another study made use of a stakeholder-centric approach that allowed election stakeholders to further understand the pros and cons of blockchain e-voting for national elections. a Blockchain-based Architecture for National E-voting System (BANES) was proposed where it made use of two vital aspects for its architecture. One is smart card technology for voter authentication and validation. The second is the zero-knowledge protocol to ensure that when a vote is cast, the blockchain determines that a valid vote has been cast, without revealing the identity of the voter or the vote information. The system made use of a blockchain built on Hyperledger Fabric framework. This was later evaluated using the Architecture Trade-off Analysis Method (ATAM). The researchers

invited top officials of the South African Independent Electoral Commission to identify key requirements crucial for an electoral system, and base their ATAM evaluation based on these requirements. Overall, experts considered that the proposed system was a plausible national e-voting system for South Africa. However, there would have to be necessary customizations to the system should it be used for other developing countries (Daramola & Thebus, 2020).

Arguments on Blockchain-based E-voting Systems

A new report from MIT by Ron Rivest, Michael Specter, Sunoo Park and Neha Narula strongly argues against the implementation of blockchain technology in e-voting on the basis that it will increase cybersecurity vulnerabilities (Powers, 2020). One of the issues is with regards to tolerance for failure. In contrast to errors encountered in online shopping or online banking, there are a number of remedies to take. As for a blockchain-based e-voting system, there is almost little to no remedy if ever a vote was altered or was not delivered to the blockchain, especially when online voting systems do not always identify when errors like these have occurred. An online voting system is also vulnerable to scalable and undetectable cybersecurity attacks, resulting in numbers of votes being exploited.

The report lays out arguments on the usage of coins as votes, the usage of a permissioned blockchain and the implementation of zero-knowledge proofs. Using coins as votes and a permissioned blockchain would suffer from key management vulnerabilities. Finally, the information verification process with the use of zero-knowledge proofs would most likely contain potential bugs and challenging cryptographic processes, and it also does not prevent physical monitoring by coercers or vote buyers. Digital processes such as this relies on various hardware and software vendors, which increases complexities and vulnerabilities to the voting process. The report notes that although blockchain technology exhibits a lot of potential, it is not recommended for online voting as it may introduce more security concerns that may compromise the voting process.

Table 1

Summary of Related Work

System/Functionalities	Smart Contracts	Smart Card Technology	Zero-Knowledge Protocol	Sidechain Technology	Ethereum	Hyperledger Fabric	Elliptic Curve Cryptography (ECC)	SHA-256 Algorithm	Shamir's Secret Sharing Algorithm	Permissioned Blockchain	Permissionless Blockchain	Feedback Mechanism	Wilson score algorithm	Points-Weighted cVoting	Protest Vote	Architecture Trade-off Analysis Method (ATAM)
Blockchain Based Online Voting System (BOVS)	✓				✓											
Blockchain-Based E-Voting System	✓				✓											
A Study on Decentralized E-Voting System Using Blockchain Technology					✓										✓	
Securing e-voting based on blockchain in P2P network							✓	✓								
Investigating performance constraints for blockchain based secure e-voting system										✓	✓					
Crypto-voting	✓			✓					✓	✓						
A Blockchain Voting System Based on the Feedback Mechanism and Wilson Score	✓				✓							✓	✓	✓		
Blockchain-based Architecture for National E-voting System (BANES)		✓	✓			✓										✓

CHAPTER 3
TECHNICAL BACKGROUND

This chapter will discuss the software tools to be used in this study.

Blockchain Technology

Blockchain is a decentralized, distributed database technology for transaction and data management that aims to create an environment where no third party is in control of transactions and data (Yli-Huumo, et.al, 2016). It is a form of Distributed Ledger Technology (DLT) that makes digital assets immutable, tamper-evident and tamper-resistant, and transparent by implementing decentralization and cryptographic hashing. It stores anonymous data in blocks chained together. As new data comes into a fresh block, it also contains a unique secure code obtained through cryptographic hashing, as well as the hash of the previous block it is chained to. Tampering with a block would result in a change in the hash of that block which would result in a mismatch of hash with its next block.

Smart Contracts

A smart contract is a self-verifying, self-executing, tamper-resistant and immutable computer program that consists of a set of rules to be run on a blockchain. It was a concept proposed by Nick Szabo in 1994. It allows a blockchain to be flexible in terms of development and design, and it is widely used to implement blockchain-based solutions to real world problems without having to use third-party systems (Mohanta, Panda, Jena, 2018). It requires transactional input before it executes its code and triggers output events. Smart contracts are mostly written in the programming language Solidity. Smart contracts will be used to program the behavior of the blockchain according to the functionality of an e-voting system.

High Level Programming Language

Solidity is an object-oriented statically-typed language for implementing smart contracts that govern the behavior of accounts within the Ethereum Virtual Machine (EVM). Solidity is the most popular language being used on the Ethereum blockchain, as well as on other private blockchains (Solidity, 2020).

Runtime Environment

Node.js is an open-source, cross-platform back-end JavaScript runtime environment that runs on Chrome's V8 JavaScript engine. It allows developers to build scalable network applications with the use of its package ecosystem, npm (Patel, 2018).

Package Manager

Node Package Manager (npm) is a command-line utility for interacting with Node.js repositories that aids in package installation, version management, and dependency management (Nodejs, 2011). It was created in 2009 as an open source project so that developers can easily share packaged modules of code. It is a command line client that allows developers to install and publish packages.

Development Framework

Truffle is a development framework for blockchains using the Ethereum Virtual Machine (EVM). It is designed to facilitate smooth development of distributed applications (DApps). Truffle integrates compilation, testing and deployment of Smart Contracts (Sahu, 2020).

Ganache is a personal blockchain for rapid Ethereum and Corda distributed application development (Ganache, n.d). It allows developers to set up a local ethereum blockchain to test Solidity-written smart contracts.

Vue is a progressive framework for building user interfaces. It is capable of powering sophisticated Single-Page Applications when used with modern tooling and supporting libraries (Vue.js, n.d).

Web Browser

Chrome is a free internet browser powered by Google since 2008 that features tabbed browsing, automatic translation, spell check of web pages and an integrated address bar (Computer Hope, 2021). It is also a browser that offers a number of browser extensions available for installation.

Browser Extension

MetaMask is a browser extension for accessing Ethereum enabled distributed applications (DApps) in a browser (Husey and Phillips, 2020). The extension injects the Ethereum web3 API into every website's javascript context, so that DApps are able to read from the blockchain. It serves as an Ethereum wallet for digital tokens that enables users to gain access to services on the blockchain network.

Performance Testing Tool

Apache JMeter is an open-source software from Apache designed to load test functional behavior and measure performance of web applications (Apache Software Foundation, 2021). It is a testing tool entirely written in Java, used to test both web and FTP applications.

Security Scanner

The Zed Attack Proxy (ZAP) is an Open Web Application Security Project (OWASP) Flagship open-source security scanner and penetration testing tool for testing web applications. (ZAP Dev Team, 2021). It is known as a "man-in-the-middle proxy" that stands between the user's browser and the web application, allowing it to intercept and inspect messages sent between browser and web application, modify the contents if needed, then forward those packets on to the destination. It can be used by both application security beginners and professional penetration testers. OWASP ZAP features include intercepting proxy servers, traditional and AJAX web crawlers, automated and passive scanners, forced browsing and scripting languages.

CHAPTER 4
DESIGN AND METHODOLOGY

This chapter highlights the operations of the research process which includes the research methods, environment and necessary instruments for data collection. This chapter also features the development procedures and tools as well as the verification and testing of the proposed systems.

4.1 Research Environment

The research was conducted on the researchers' local computers. Both the BEVS and CEVS were deployed and tested locally on different database environments to assess the performance and security of the systems.

4.2 Research Instrument and Sources of Data

The researchers gathered data on the performance of the systems by conducting automated testing using software testing tools, specifically Apache JMeter for the evaluation of the CEVS running on both a local and cloud database. The assessment methods for the BEVS running on both a local and cloud database are researcher-made with the help of Truffle and JavaScript scripting since the availability of the testing tools for the given system are limited considering blockchain is an emerging technology. To evaluate the security of the systems, the researchers manually conducted vulnerability assessment using OWASP ZAP, a security scanner, for both the CEVS and BEVS .

4.3 Research Procedure

4.3.1 Gathering of Data

Data is gathered from the testing process that measures the performance of the CEVS and the BEVS using performance testing tools, particularly Apache Jmeter with the integration of Selenium WebDriver for the CEVS, and Truffle and JavaScript for the BEVS. Both systems were

tested in two environments with respect to the nature of its database - local and cloud. The testing process is in the form of an automated simulation where first, a user sends a request to login to the system. Upon successful login, the user selects candidates, then submits their ballot for review and confirms their ballots, where a ballot receipt will also be displayed. Finally, the user is redirected to view the election results. These events are done simultaneously within a specific period of time to simulate normal to heavy loads on the system. The data is then analyzed based on performance metrics, such as the average response time and average error rate. Also included in the performance test is determining the amount of load the system can handle before the system becomes unresponsive.

Security testing is conducted with the use of OWASP ZAP, an open-source web application security scanner. OWASP ZAP's Automated Scan tool passively scans through the system and finds pages within a given URL. With the list of discovered pages, it proceeds to automatically attack functionalities and parameters with its active scanner. This process will generate a list of alerts with their corresponding risk levels, as well as the different instances associated with each of the alerts found in the system. Data gathered from this assessment are then analyzed and compared with the two systems.

4.3.2 Treatment of Data

Gathered data from the testing simulations are tabulated for presentation, analysis and interpretation. Data gathered from the performance testing tools are used to measure and calculate for the average response time, average error rate and the breaking point of each system. Data gathered from the vulnerability assessment are used to identify the vulnerabilities present in each of the systems, and these will be analyzed as to which vulnerabilities are of high risk to the security of the systems and how these can be mitigated.

4.4 Concept

The research concept is that of an empirical analysis of an e-voting system using blockchain technology particularly in terms of performance and security. The research used the Ganache software application as the blockchain network for the BEVS. Both systems for testing and evaluation are automated using automated software testing tools. The environments for evaluating the performance and security of the systems were conducted in local and cloud environments.

4.4.1 Conceptual Framework

Both systems were designed to have identical user interfaces. The difference between the two systems lies in the implementation of the backend development – the BEVS uses Ganache as its local blockchain network and the Ropsten Ethereum as its cloud blockchain network, while the CEVS uses MongoDB as its database system both for local and cloud. Both systems consist of the three modules, namely the Login, Voting, and Results Modules. The Login Module is implemented differently on the systems. Users in the CEVS will use their respecting voter's ID in accessing the voting application, while users in the BEVS will import their digital wallet MetaMask, and login by connecting to the blockchain network. In the Voting Module, users are shown the list of candidates they can cast their vote on, and they can verify their selection with a ballot receipt. Vote processing is done after a user confirms their ballot receipt in the CEVS or issues a vote transaction in the BEVS. Users will be redirected to view the election results in the Results module.

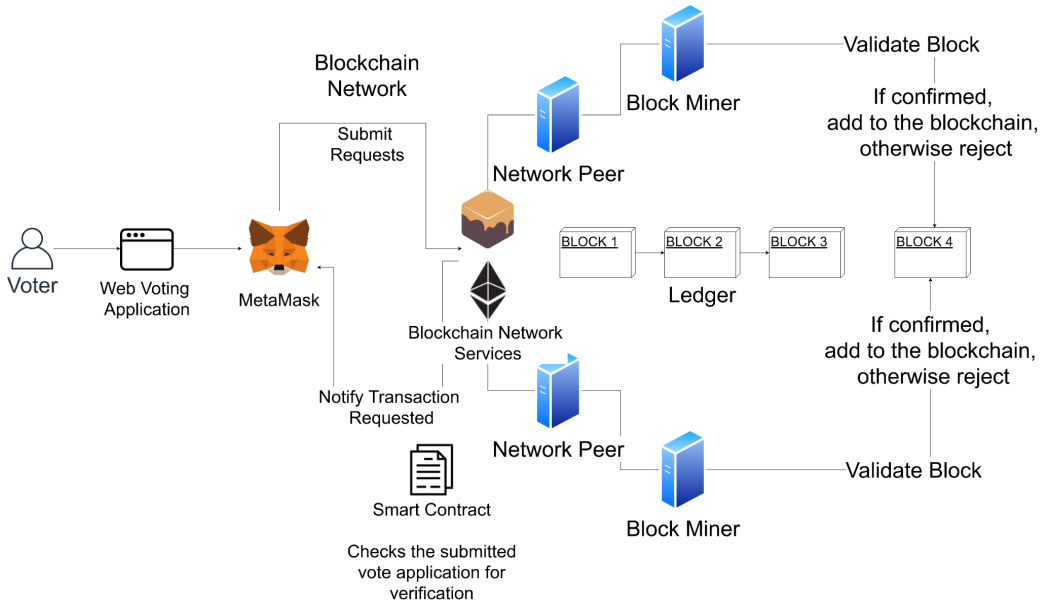


Figure 1. BEVS Conceptual Framework

In the BEVS, users in the system will use the private key of their digital wallet as their login credentials to be imported to MetaMask and connect it to the network. The MetaMask extension is required to handle the requests from the user in submitting their votes to Ganache and the Ropsten network. The smart contracts programmed in the system determine whether the request meets the terms and conditions set in the contract. If all conditions are met, a new block will be created, processed and then verified to be added to the ledger of the blockchain network that serves as the database of the system.

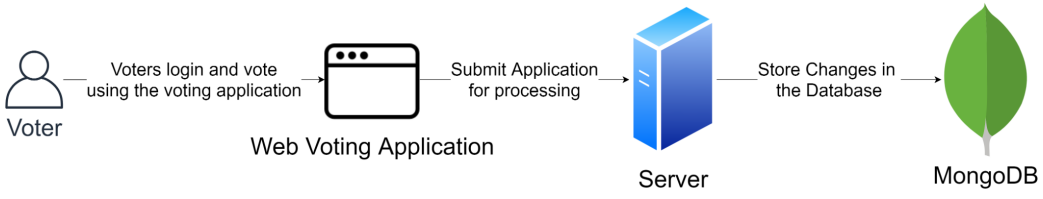


Figure 2. CEVS Vote Processing Conceptual Framework

In the CEVS, users are required to login using their voter ID number. Casted votes will be processed by the server, and are stored in a database that automatically tallies the vote counts of each candidate.

4.5 Analysis and Design

Electronic voting systems are software platforms that are used to administer votes and elections in a safe manner. They remove the need to vote on paper or gather in person because they are a digital forum. They also preserve the legitimacy of your vote by making it impossible for voters to vote more than once. Many secure voting software providers provide vote control consultancy services to assist organizations with the creation and implementation of voting procedures. These services assist organizations in saving time, adhering to best practices, and meeting internal and/or external criteria, such as third-party vote administration requirements.

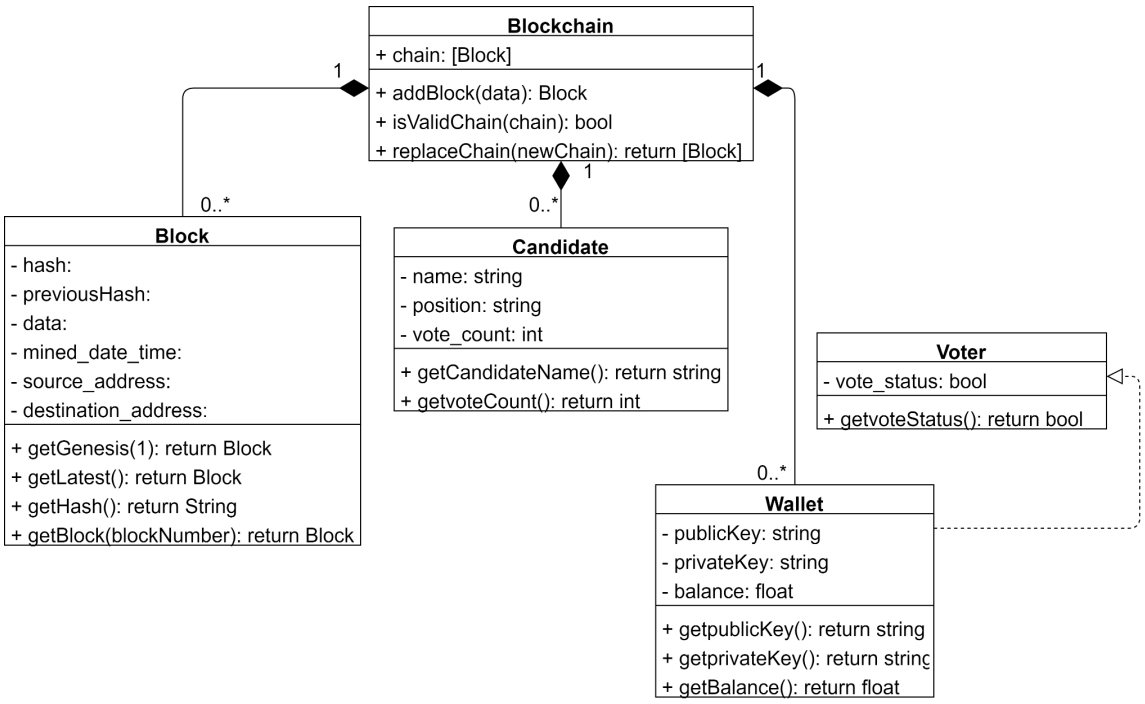


Figure 3. UML Diagram for BEVS

The BEVS consists of a block, wallet, voter, and candidate as shown in Figure 4. Each submission of a vote is considered to be a new block in the blockchain. The blockchain must have all blocks verified for validity in order for the chain to be valid. Invalid blocks are discarded from the blockchain. Each voter must have a digital wallet to be eligible for voting. For the candidate, a “vote_count” is provided to keep track of the number of votes the candidate receives by the voters. The voter will have the ability to cast votes which will then be verified before being added to the blockchain.

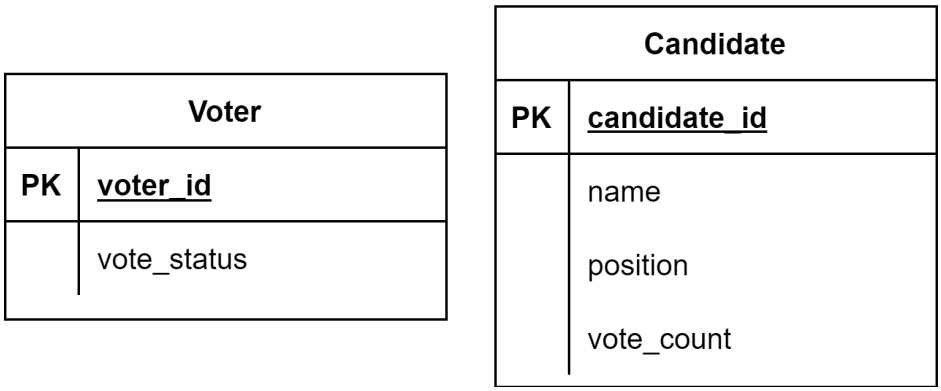


Figure 4. ERD for CEVS

The CEVS consists only of a voter and candidate. Figure 5 shows that a voter must be registered on the database in order to access the voting system by using the voter id. Each voter has a “vote_status” to indicate whether a voter has voted or not. For the candidate, a “vote_count” is provided to keep track of the number of votes the candidate receives by the voters.

4.6 Development Model

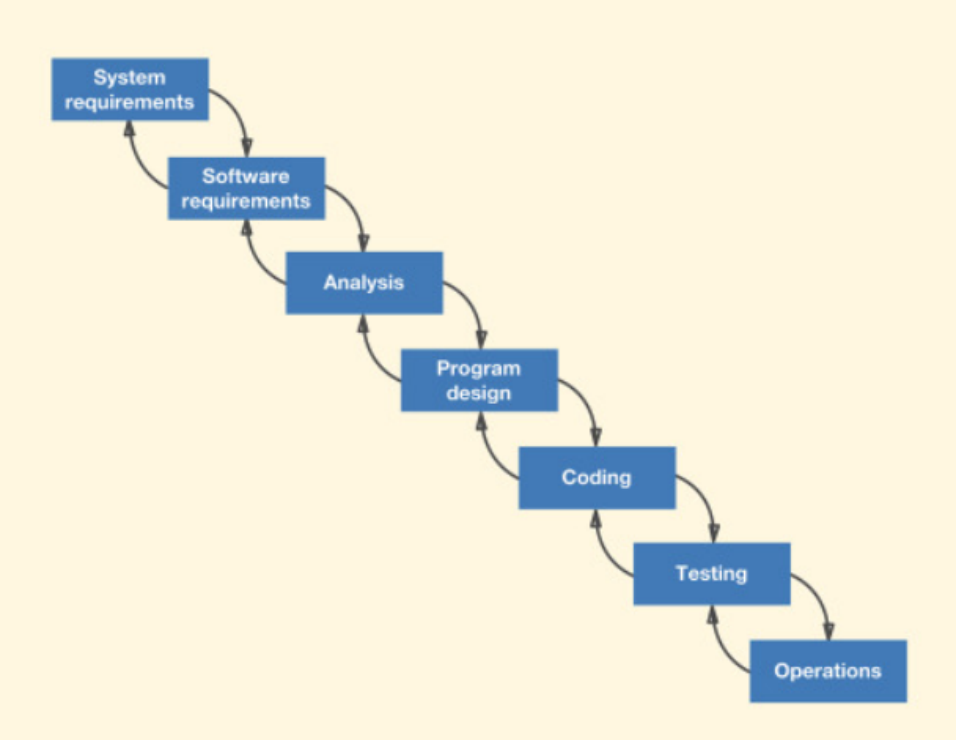


Figure 5. Modified Waterfall Model

The researchers followed the Modified Waterfall Model for the development of the system in this study. This allows the researchers to go back to the previous step if there were to be necessary changes identified in each phase.

System Requirements: In this phase, the researchers researched on their selected topic and identified the objectives of this study. The researchers identified the system requirements that were essential in achieving the objectives of this study.

Software Requirements: This phase entails the research and identification of software tools and applications used in the development of the system.

Analysis: This phase entails the formulation of research methodologies in the study. It will also include the identification of criteria or metrics for the

evaluation of the identified system and software requirements before proceeding to the actual development of the system.

Program Design: This phase entails defining the system architecture, database design, functional modules and interfaces of each system.

Coding: This is the phase wherein the researchers used the necessary tools and applications in creating the systems. The program design is used in this phase as a basis for the development of the system.

Testing: The researchers tested the system to check if all modules are working properly and if there are no more bugs to be fixed before proceeding to the next phase.

Operations: The systems are deployed on the local and cloud so that the researchers can have a different controlled environment to conduct tests for the performance and security of the system. This phase will also involve the analysis and evaluation of results gathered from the automated testing.

4.7 Development Approaches

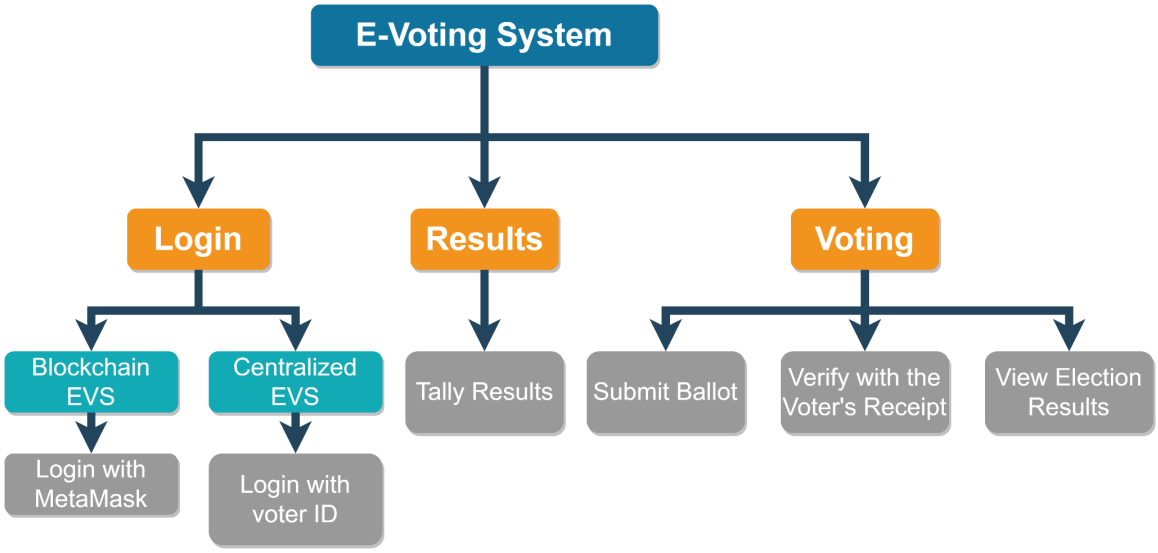


Figure 6. Top-Down Development Approach of E-Voting System

Figure 7 shows the Top-Down development approach which is utilized in the development of the proposed e-voting systems. Both the BEVS and the CEVS have the same modules and similar interfaces, but their implementations are different. The systems are divided into subsystems according to each functionality or module. Sufficient planning on the design and implementation, as well as further research on the software tools used are necessary before beginning the initial development. This is to ensure that the requirements of the system are clearly identified and that the system is appropriately developed according to these requirements.

4.8 Software Development Tools

The table below shows the tools, along with their respective versions, that the researchers used in developing the systems.

Table 2

List of Software Tools

Software	Version	Use
Node.js	14.16.0	Server development
Node Package Manager (npm)	7.6.3	Package manager for Node.js codes
Truffle	5.2.5	Development environment for Ethereum
Ganache	6.12.2	Local blockchain for Ethereum distributed application development
MetaMask	9.2.0	Browser extension for accessing Ethereum DApps
Vue.js	3.0.7	Development framework
MongoDB	4.4.5	Non-relational database management system
Visual Studio Code	1.54	Text editor for web development
Google Chrome	89.0.4389.90	Browser to run the application on
Apache JMeter	5.4.1	Performance testing tool
OWASP ZAP	2.10.0	Security scanner

4.9 Project Management

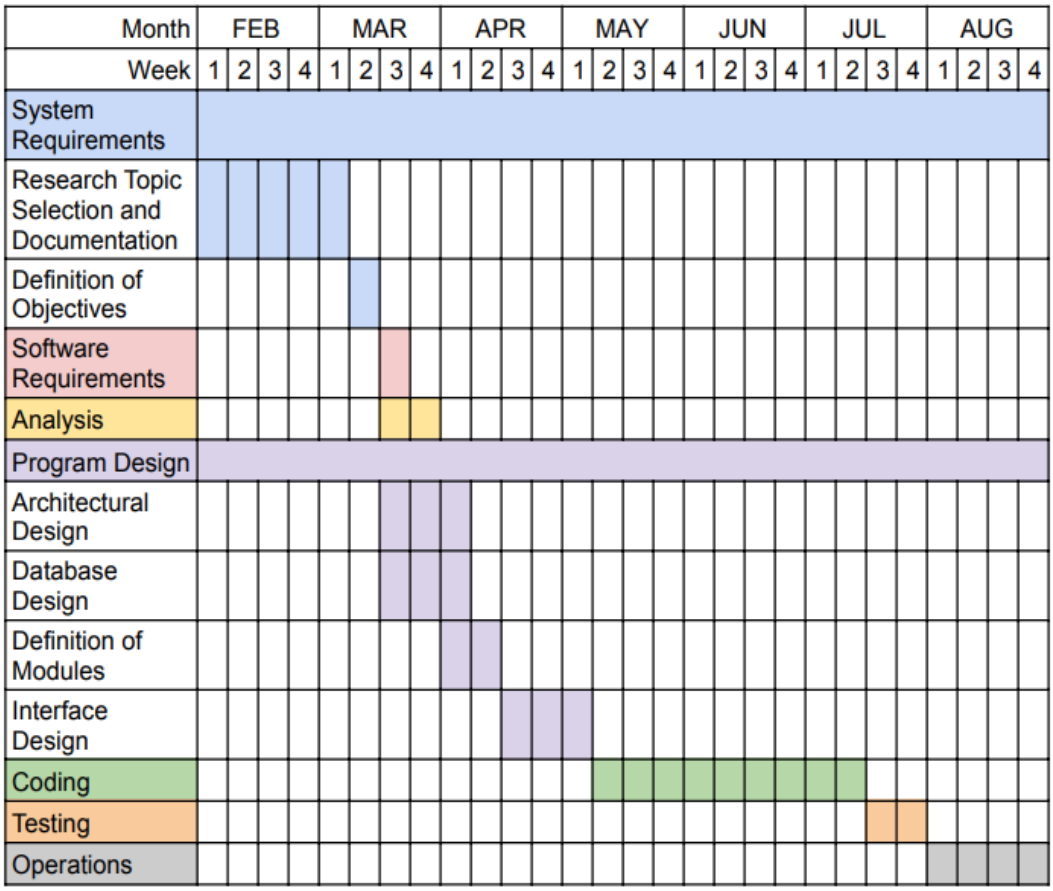
This section of the study includes subsections, particularly Schedule and Timeline, Responsibilities, and Budget and Cost Management.

4.9.1 Schedule and Timeline

The researchers will follow the tabulated schedule and timeline for the research procedure and system development.

Table 3

Gantt Chart of Activities



4.9.2 Responsibilities

In the research methodology and system development, the researchers will equally share all of the tasks. The researchers will be cooperating with each other to finish their responsibilities following the aforementioned schedule and timeline.

Table 4

Roles and Responsibilities

Member	Role	Responsibility
Jayson V. Cadiz	Researcher	Gather data from respondents
	Developer	Research on software tools and applications
		Develop both systems
		Test the developed systems
Nicole Amber M. Mariscal	Researcher	Gather data from respondents
	Developer	Research on software tools and applications
		Develop both systems
		Test the developed systems

4.9.3 Budget and Cost Management

The researchers will follow the estimated budget and cost management tabulated below.

Table 5

Proposed Budget and Cost Management

Items	Cost
Personal Desktops	Php 80,000.00
Transportation Fees	Php 1,000.00
Conference Fees	Php 5,000.00
Miscellaneous Fees	Php 5,000.00
Total	**Php 91,000.00**

4.10 Verification, Validation and Testing

The proposed systems will be tested by the researchers using different testing tools, particularly performance testing tools to evaluate the performance of the systems, and vulnerability assessment tools and penetration testing tools to evaluate the security of the systems.

To test the security of the BEVS and the CEVS, vulnerability assessment will be done using vulnerability assessment tools and penetration testing tools. This will identify the vulnerabilities of each system and determine if these vulnerabilities are critical threats to the overall security of the system.

On the other hand, the performance of the BEVS and CEVS will be analyzed and compared with one other based on these performance metrics.

The following are the performance metrics in assessing the systems performance:

Average Response Time: This is a direct measure of the average of the total amount of time the system takes to respond to a request sent by the client within a given time period.

$$ART = \frac{\sum(time\ of\ response - time\ on\ input)}{number\ of\ responses\ in\ selected\ time\ period}$$

Average Error Rate: This is a calculation that generates the average percentage of failed requests compared to the total number of requests. The percentage reflects HTTP status codes indicating an error on the server, as well as any request that times out before completing its response.

$$AER = \sum\left(\frac{failed\ events}{successful\ events + failed\ events}\right) \times 100$$

Breaking Point: This is used to determine the maximum load-bearing capacity of the system. Several automated requests will be sent to the system, simulating many active virtual users, and the breaking point of the system will be measured as the number of requests at which the system crashes.

CHAPTER 5
RESULTS AND ANALYSIS

This chapter contains results gathered from the development of the systems and conducting performance and security tests on the systems based on the objectives presented in Chapter 1.

5.1 CEVS Behind the Process

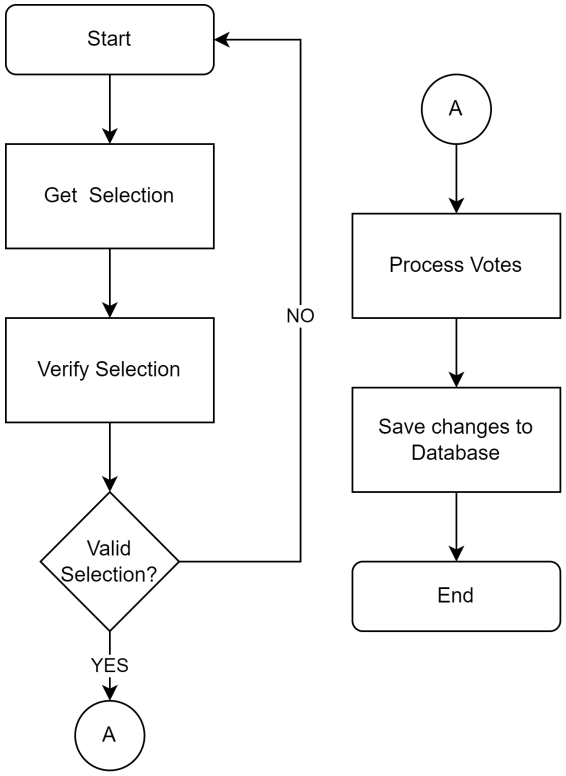


Figure 7. Behind the Process of CEVS Flowchart

The traditional e-voting system uses the common centralized process wherein a voter's voting application is managed by a server upon request. These processes are from performing validation proceedings and other necessary processings that need to be performed to completely process requests. In the context of e-voting systems, when one submits a voting application, it first verifies the selection then processes the selection to be recorded into votes and finally counts the votes to the respective candidates and save it on the database.

5.2 BEVS Behind the Process

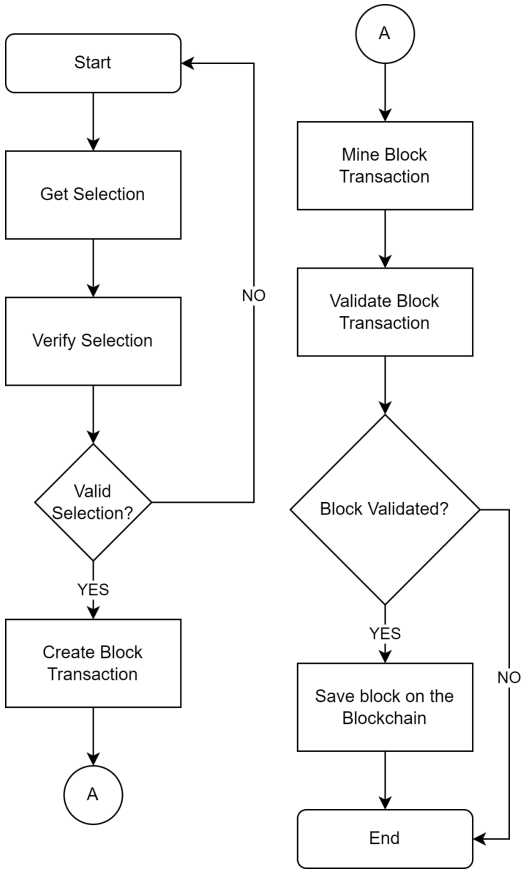


Figure 8. Behind the Process BEVS Flowchart

The implemented e-voting system solution uses the blockchain technology to process requests wherein a voter's voting application is managed by more than one upon request. The concept of processing votes is similar to the CEVS but the only difference is when a request takes place, a transaction will be prompted to make changes on the blockchain network. These created transactions are saved on a block to be mine, verified, and be validated only if the rules on the smart contract are met. Otherwise, the transaction will be invalid. If the circumstances make the block valid, then the block will be saved on the blockchain network.

5.2.1 Smart Contract

Table 6

Election Smart Contract

Election			
	Data Type	Variables	Task
Candidate Model	uint	id	Candidate id number
	string	name	Candidate name
	string	position	Candidate position
	uint	vote_count	Candidate vote count
	mapping	voters	Record voters account
	mapping	candidates	Records list of candidates
Functions			
Name		Task	
addCandidate		Inserts a candidate to the candidate list	
vote		Prompts a transaction to make changes on the blockchain	
getVotedAcc		Checks if an account has voted or not	
getCandidates		Retrieves the list of candidates	

The smart contract of the blockchain network is the main key factor in the overall structure of the network. The set functions and variables

dictate what the blockchain can use and can do whenever the contract is called to make requests.

5.3 Performance Testing

The performance test was conducted in two different environments – both systems were run with a local database and a cloud database. Using Apache JMeter with the integration of Selenium WebDriver extension for the testing scripts, all tests were run on a sample of 50 threads in a time period of 250 seconds, with a new thread executing every 5 seconds. Each test done on the CEVS is composed of 50 threads, simulating 50 virtual users, while in the BEVS, each test simulates 50 different blockchain accounts simultaneously performing a transaction on the blockchain network. These events have been automated to record average response times and average error rates of modules. For the Login Modules of both systems, the methods on performing the login process are distinct from each other. The CEVS checks for existing voter ID on the database while the BEVS only checks if MetaMask is connected to the server without prompting any requests to the blockchain network. Hence, recording the performance of the Login Module is irrelevant to the discussion. The breaking point performance test is done by constantly increasing the amount of simultaneous threads for the CEVS and transactions for the BEVS.

5.3.1 Performance Testing on a Local Database

This subsection presents the performance test results of running the CEVS on a local MongoDB database and the BEVS on Ganache, a local private Ethereum blockchain. Performance data are gathered through analyzing the average response time of submitted ballots in the Voting Module and average response time of the retrieval of results in the Results Module, as well as the average error rates per module and breaking points of both systems.

Table 7 presents the performance test results for the Voting Module of the two systems. Each test simulates 50 virtual users casting their votes in the Voting Module. The average response time recorded is the average

of the time from when a user reviews their votes in the Voting Module until the time a ballot receipt is issued, signifying a successful transaction of votes. Although the CEVS processes requests faster, the BEVS shows promise with its consistent 0.00% error rate despite its slightly slower response time.

Table 7

Performance Test Results on the Local Voting Module

Tests	Average Response Time		Error Rate	
	BEVS	CEVS	BEVS	CEVS
T1	1.745 seconds	0.077 seconds	0.00%	2.00%
T2	1.683 seconds	0.071 seconds	0.00%	0.00%
T3	1.780 seconds	0.056 seconds	0.00%	0.00%
T4	1.885 seconds	0.035 seconds	0.00%	2.00%
T5	1.705 seconds	0.055 seconds	0.00%	6.00%
T6	1.822 seconds	0.289 seconds	0.00%	10.00%
T7	1.694 seconds	0.030 seconds	0.00%	2.00%
T8	1.649 seconds	0.049 seconds	0.00%	2.00%
T9	1.910 seconds	0.168 seconds	0.00%	4.00%
T10	1.684 seconds	0.213 seconds	0.00%	8.00%

Table 8 presents the performance test results for the Results Module of the two systems. The average response time monitored is the time from when the server completes vote processing to the time the updated results are shown to the user. Although the page instantly loads and displays page content, the actual count of votes takes a while to appear on the charts. This highlights how long it takes to retrieve data

from the server. There is only a slight difference in the average times of both systems.

Table 8

Performance Test Results on the Local Results Module

Tests	Average Response Time		Error Rate	
	BEVS	CEVS	BEVS	CEVS
T1	0.609 seconds	0.644 seconds	0.00%	0.00%
T2	0.638 seconds	0.885 seconds	0.00%	0.00%
T3	0.831 seconds	0.841 seconds	0.00%	0.00%
T4	0.650 seconds	0.690 seconds	0.00%	0.00%
T5	0.618 seconds	0.760 seconds	0.00%	0.00%
T6	0.645 seconds	0.923 seconds	0.00%	0.00%
T7	0.591 seconds	0.619 seconds	0.00%	0.00%
T8	0.598 seconds	0.643 seconds	0.00%	0.00%
T9	0.660 seconds	1.034 seconds	0.00%	0.00%
T10	0.544 seconds	1.510 seconds	0.00%	0.00%

Table 9 presents the average error rates of each module in the two systems with a local database. The BEVS shows promise having no errors as compared to the CEVS. The errors that occurred in this CEVS Voting Module were logged with an HTTP status code of 500 – Internal Server Error. These errors are requests of vote processing that have failed on the server side and are taken care through error handling with the aim to enable the application to proceed despite manifesting errors. In the event of these errors, the casted votes of that account were not counted, thus bearing no change to the election results.

Table 9

Average Error Rates in Local Database Environment

System Modules	BEVS	CEVS
Login Module	0.00%	0.00%
Voting Module	0.00%	3.6%
Results Module	0.00%	0.00%

Upon further observation of both systems for their breaking points, the CEVS shows no proof of the system crashing. However, there were multiple error occurrences in the entire duration of the testing process, but these errors did not pose any vital issues that could disrupt other requests given to the system. The system handled any error occurrences, thus allowing the system to continue to perform other processes and prevent the system from becoming completely unresponsive. Similarly, the BEVS also exhibits no manifestations of the system crashing. The system barely processes requests due to high system load. Nonetheless, all requests were fulfilled successfully despite having performance response times affected.

5.3.2 Performance Testing on a Cloud Database

This subsection presents the performance test results of running the CEVS on a cloud database using MongoDB Atlas and the BEVS on Ropsten, a public blockchain test network. Testing is done similar to the procedures in the testing of the systems with a local database.

Table 10 presents the test results for both systems' Voting Module. Similar to the test results in the local database environment, the CEVS processes votes faster. The BEVS running on a public blockchain network performs extremely slow as transactions in the public blockchain network are processed in a block together with other transactions in the network

before they are validated. Despite its slower response time, the BEVS shows promise having encountered no errors in all of the tests conducted.

Table 10

Performance Test Results on the Cloud Voting Module

Tests	Average Response Time		Error Rate	
	BEVS	CEVS	BEVS	CEVS
T1	50.7 seconds	1.683 seconds	0.00%	16.00%
T2	19 seconds	1.701 seconds	0.00%	24.00%
T3	17.9 seconds	1.862 seconds	0.00%	14.00%
T4	28.7 seconds	1.908 seconds	0.00%	22.00%
T5	17.3 seconds	1.681 seconds	0.00%	14.00%
T6	21.2 seconds	1.727 seconds	0.00%	8.00%
T7	17.7 seconds	1.943 seconds	0.00%	12.00%
T8	17.9 seconds	1.822 seconds	0.00%	18.00%
T9	15.4 seconds	1.754 seconds	0.00%	16.00%
T10	19.4 seconds	1.706 seconds	0.00%	20.00%

Table 11 presents the test results for both systems' Results Module. Similar to the test results in the local database environment, there is only a slight difference in the average times of both systems, with the CEVS having the fastest average response time. However, the BEVS fares better having no errors compared to the CEVS with constant server errors in each test.

Table 11

Performance Test Results on the Cloud Results Module

Tests	Average Response Time		Error Rate	
	BEVS	CEVS	BEVS	CEVS
T1	1.495 seconds	1.270 seconds	0.00%	12.00%
T2	1.466 seconds	1.257 seconds	0.00%	12.00%
T3	1.456 seconds	1.472 seconds	0.00%	10.00%
T4	1.404 seconds	1.362 seconds	0.00%	12.00%
T5	1.376 seconds	1.376 seconds	0.00%	2.00%
T6	1.356 seconds	1.296 seconds	0.00%	4.00%
T7	1.387 seconds	1.292 seconds	0.00%	10.00%
T8	1.377 seconds	1.369 seconds	0.00%	4.00%
T9	1.367 seconds	1.243 seconds	0.00%	8.00%
T10	1.413 seconds	1.598 seconds	0.00%	12.00%

Table 12 presents the average error rates of each module in the two systems with a cloud database. Similar to the tests conducted on the systems with a local database, the BEVS exhibited no errors compared to the CEVS, which exhibited a higher percentage of errors in all of its modules than the CEVS running a local database. This proves that the CEVS, both on a local and cloud database, is more prone to server errors, specifically in the Voting Module, that is a core component in an e-voting system.

Table 12

Average Error Rates in Cloud Database Environment

System Modules	BEVS	CEVS
Login Module	0.00%	7.4%
Voting Module	0.00%	16.4%
Results Module	0.00%	8.6%

Upon further observation of both systems for their breaking points, the CEVS shows a point wherein requests were no longer processed due to the constraints on the database. The server is limited to a number of 500 connections only. Thus, the server can handle only the given amount of connections before the system becomes completely unresponsive. The BEVS, on the other hand, can accommodate as many requests as possible. The only occurrence that the system rejects requests is when a single account is prompting too many requests at the same time, and with this occurrence, the system can still process other requests without the system crashing.

5.4 Security Testing

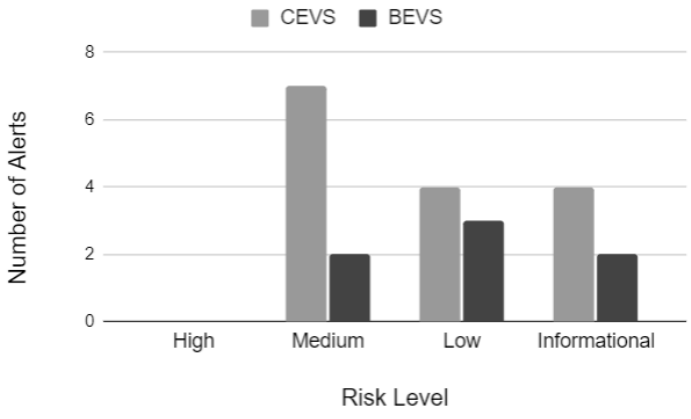


Figure 9. Alerts and Their Risk Levels

The security of the two systems were tested with the use of OWASP ZAP's Automated Scan tool. Figure 10 shows the number of alerts with their corresponding risk levels found on the two systems. As shown in Figure 10, there is a higher number of alerts found in the CEVS, with most alerts of a medium risk level. Note that some alerts found in the CEVS are listed twice, as these alerts were detected on both the server and the frontend application during the scanning procedure.

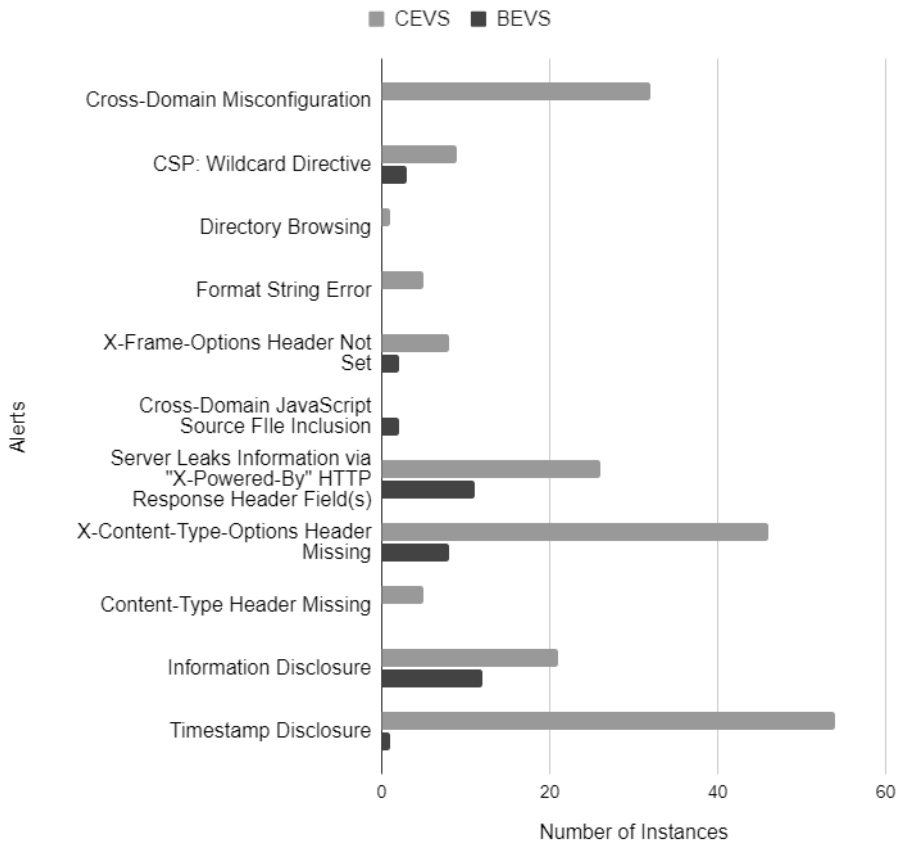


Figure 11. Alerts and Their Instances

Figure 11 shows the different instances of the alerts, arranged by risk level from Medium to Informational. There are more vulnerabilities found in the CEVS compared to the BEVS. Although most of the alerts were detected in both systems, there is a significantly higher number of instances found in the CEVS.

Upon further research into the alerts generated by the security scan, the risks associated with each found alert are listed in Table 13.

Table 13

Alerts and Their Risks

Alerts	Risks
Cross-Domain Misconfiguration	CORS-based attacks
CSP: Wildcard Directive	Cross-site Scripting (XXS) and data injection attacks
Directory Browsing	Brute force attacks
Format String Error	Format string exploits
X-Frame-Options Header Not Set	Clickjacking attacks
Server Leaks Information via “X-Powered-By” HTTP Response Header Field(s)	Exploits on frameworks and components of a web application
X-Content-Type-Options Header Missing	Cross-site Scripting (XXS) attacks
Content-Type Header Missing	Cross-site Scripting (XXS) attacks
Information Disclosure	Suspicious comments may help an attacker understand the underlying application logic and possible weakness, thus allowing an attacker to create a malicious attack.
Timestamp Disclosure	Sensitive information retrieval

CHAPTER 6
CONCLUSION AND RECOMMENDATION

This section elaborates the findings of the conducted research in performing the testing and evaluation of the BEVS and CEVS as well as addressing the significance of an e-voting system with the emerging blockchain technology. The recommendations are elaborated with regards to the implemented e-voting systems in this research, as to how this research can be conducted in different perspectives.

An analysis has been made on the implementation of blockchain technology on e-voting systems. This is done by developing a centralized e-voting system and a blockchain-based e-voting system, and testing both systems to scale and compare their performance and security. The concluding observations from this research are listed below.

- The performance of the BEVS is better with the use of a local blockchain network than that of a public blockchain network.
- The performance of the BEVS is slower by at least 1 second compared to the CEVS. However, the BEVS is more reliable given its lower error rates.
- The BEVS is more secure given the few vulnerabilities that were found in the security scan as compared to the CEVS.

Future researchers should conduct testing in a live controlled group for a more practical and realistic approach. Future studies can also implement e-voting systems for a bigger scale of users, possibly simulating a nationwide voting scenario, and investigate its scalability. Security evaluations could also be done in-depth, such as specifically targeting the OSI layers of the network. Another recommendation is to make use of other consensus algorithms and blockchain types that best fit the requirements of the system.

GLOSSARY

Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a network.

BEVS - Blockchain E-Voting System is an e-voting system that uses the blockchain technology approach

CEVS - Centralized E-Voting System is an e-voting system that uses the centralized approach

Cybersecurity is the protection from cyber threats by the use of technologies, processes, and policies to safeguard systems, networks, programs, devices, and data Cyber attacks will be reduced, and systems, networks, and technology will be protected from unlawful use.

Distributed Systems are a collection of independent components located on different machines that share messages with one another in order to achieve common goals.

E-Voting Systems are systems that utilize electronic means to cast and count votes.

Performance is the ability of a system to perform actions given certain conditions

BIBLIOGRAPHY

Journal Article

Abu-Shanab, E., Knight, M. & Refai, H. (2010). E-voting systems: A tool for e-democracy. *Management Research and Practice.* 2. 264-274.

Atlam, F., Wills, G. (2019). Chapter Three - Intersections between IoT and distributed ledger. *Advances in Computers. 115.* 73-113. https://doi.org/10.1016/bs.adcom.2018.12.001

Yi, O.K. & Das, D. (2019). Blockchain Technology For Electronic Voting. *Journal of Critical Reviews, 7*(3), 114-124. http://dx.doi.org/10.31838/jcr.07.03.22

Patil,H., Rathi, K. & Tribhuwan, M. (2018). A Study on Decentralized E-Voting System Using Blockchain Technology. *International Research Journal of Engineering and Technology (IRJET), 5*(11). 48-53.

Yi, H. (2019). Securing e-voting based on blockchain in P2P network. *EURASIP Journal on Wireless Communications and Networking,* 1-9. https://doi.org/10.1186/s131638-019-1473-6

Ma, X., Zhou, J., Yang, X. & Liu, G. (2020). A Blockchain Voting System Based on the Feedback Mechanism and Wilson Score. *Information*, *11*(12), https://doi.org/10.3390/info11120552

Daramola, O. & Thebus, D. (2020). Architecture-Centric Evaluation of Blockchain-Based Smart Contract E-Voting for National Elections. *Informatics, 7*(16), 1-22. https://doi.org/10.3390/informatics7020016

Khan, K., Arshad, J. & Khan, M. (2019). Investigating performance constraints for blockchain based secure e-voting system. *Future Generation Computer Systems, 105*, 13-26. https://doi.org/10.1016/j.future.2019.11.005

Hjálmarsson, F., Hreiðarsson, G., Hamdaqa, M. & Hjálmtýsson, G. (2018) Blockchain-Based E-Voting System, *IEEE 11th International Conference on Cloud Computing,* 983-986, doi: 10.1109/CLOUD.2018.00151.

Yli-Huumo, J., Ko, D., Choi, S., Park, S. & Smolander, K. (2016). Where Is Current Research on Blockchain Technology?—A Systematic Review. PLoS ONE 11(10): e0163477. doi:10.1371/journal. Pone.0163477

Lin, I., Liao, T. (2017). A Survey of Blockchain Security Issues and Challenges. *International Journal of Network Security. 19*(5), 653-659, DOI: : 10.6633/IJNS.201709.19(5).01

Shrivas, M. K., & Dr. Yeboah, T. (2018). The Disruptive Blockchain: Types, Platforms and Applications. *Texila International Journal of Academic Research (TIJAR)*, Special Edition(2019), 17-39. doi:DOI: 10.21522/TIJAR.2014.SE.19.02.Art003

Web Article

Chi, L. (2016). Philippine elections hack 'leaks voter data'. *BBC News*. https://www.bbc.com/news/technology-36013713

Espina, M. (2019). Election 2019 technical issues: Paper jams, malfunctioning machines. *Rappler*. https://www.rappler.com/nation/elections/230425-vote-counting-machines-errors-technical-issues/

USC COMELEC Investigation Committee. (2021). COMELEC investigation committee report on the failure of the 2021 USC-SSC elections. *Facebook.* https://web.facebook.com/usccomelec/posts/4013553535349177

Webster, P. (2021). First as Tragedy then as Farce: What Happened to the 2021 USC-SSC Elections?. *Facebook.* https://web.facebook.com/todayscarolinian/posts/10159360219518814

Powers, B. (2020). New MIT Paper Rejects Blockchain-Based Voting Systems. *CoinDesk.* https://www.coindesk.com/mit-paper-rejects-blockchain-based-voting-systems-elections

Eulau, H. , Gibbins, R. and Webb, P. (2020). Election. *Encyclopedia Britannica*. https://www.britannica.com/topic/election-political-science

Kuca, G. (2019). Secret ballot. *Max Planck Encyclopedia of Comparative Constitutional Law.* https://oxcon.ouplaw.com/view/10.1093/law-mpeccol/law-mpeccol-e683

Seal, A. (2020). Centralized vs Decentralized Network: Which One Do You Need? *vXchnge*. https://www.vxchnge.com/blog/centralized-decentralized-network

Touron, M. (2019). Centralized vs Decentralized vs Distributed Systems. *Berty.* https://berty.tech/blog/decentralized-distributed-centralized/

Peralta, R. (2016, May 23). Electronic voting. *Encyclopedia Britannica*. https://www.britannica.com/topic/electronic-voting

Solidity. (2020). Solidity. *Solidity.* https://docs.soliditylang.org/en/v0.6.8/

Patel, P. (2018). What exactly is NodeJS?. *freeCodeCamp.*

https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/

Nodejs. (2011). What is npm? *Nodejs.*
https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/

Sahu, M. (2020). *What is Truffle Suite? Features, How to Install, How to Run Smart Contracts*. upGrad Blog https://www.upgrad.com/blog/what-is-truffle-suite/

Ganache. (n.d). Ganache Overview. https://www.trufflesuite.com/docs/ganache/overview

Vue.js. (n.d). Introduction. https://vuejs.org/v2/guide/

Computer Hope. (2021). Chrome. https://www.computerhope.com/jargon/c/chrome.htm#overview

Hussey, M. and Phillips, D. (2020). MetaMask: What It Is and How To Use It. *Decrypt.* https://decrypt.co/resources/metamask

Apache Software Foundation. (2021). Apache JMeter. https://jmeter.apache.org/index.html

ZAP Dev Team (2021). Introducing ZAP. https://www.zaproxy.org/getting-started/

Research Paper

D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun, A review on consensus algorithm of blockchain, 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017, pp. 2567-2572, doi: 10.1109/SMC.2017.8123011.

Fusco, F., Lunesu, M., Pani, F. & Pinna, A. (2018). Crypto-voting, a Blockchain based e-Voting System. Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 3, 223-227. DOI: 10.5220/0006962102230227.

Krimmer, R. (2012). The Evolution of E-voting: Why Voting Technology is Used and How it Affects Democracy.

Ferdous, Chowdhury, Hoque and Colman. (2020). Blockchain Consensus Algorithms: A Survey. doi: arXiv:2001.07091

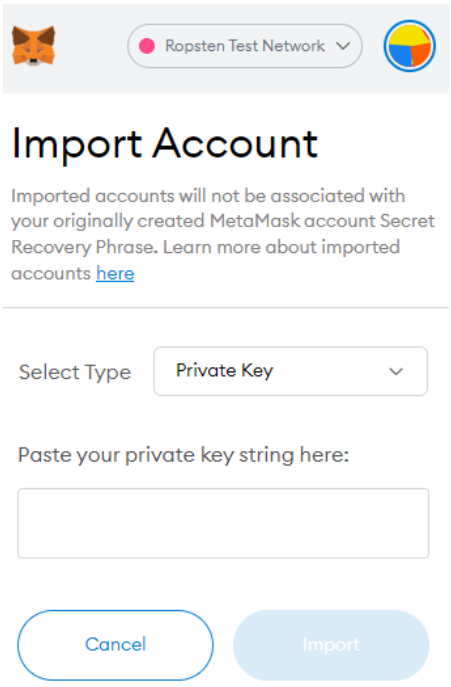
Endemann, B., Wladawsky-Berger, I., LaPointe, C., Yen, H. (2020). Blockchain. Technology Factsheet Series.

Mohanta, B., Panda, S., & Jena, D. (2018). An Overview of Smart Contract and Use Cases in Blockchain Technology. 10.1109/ICCCNT.2018.8494045

APPENDICES

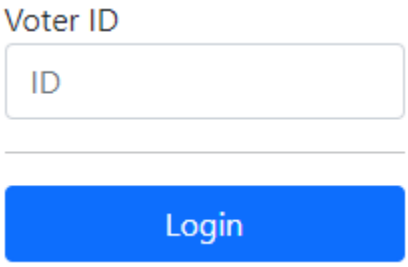
APPENDIX A

System Requirement Specifications



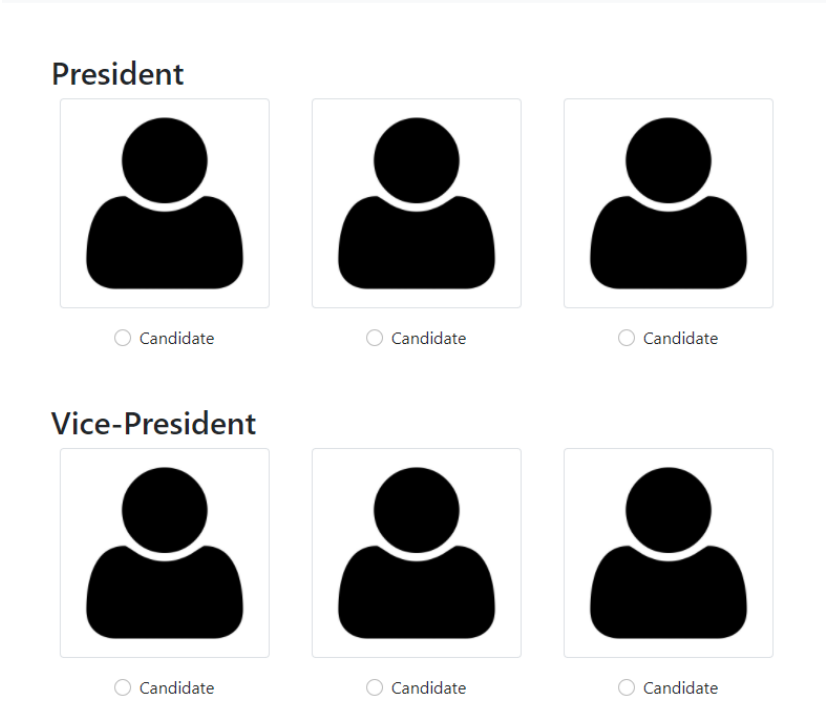
BEVS Login Module

The image above shows the login procedure for the BEVS. It requires the user to import a private key of their wallet to be able to proceed to the voting application.



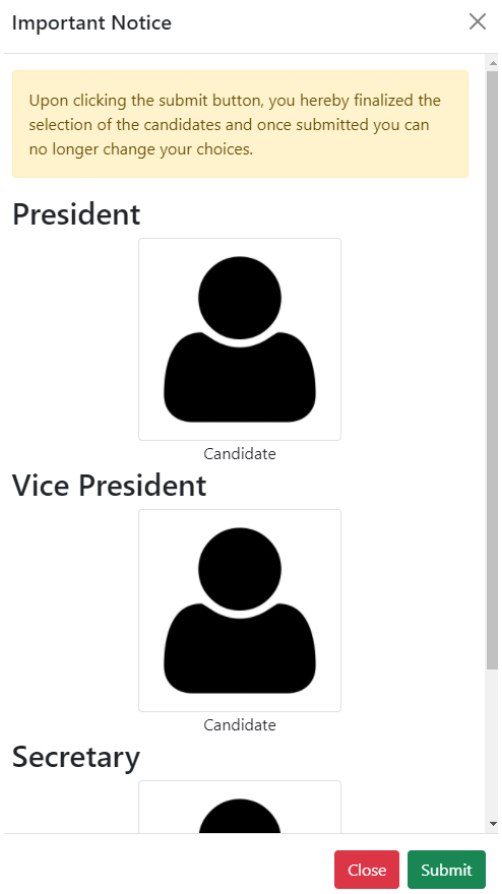
CEVS Login Module

The image above shows a login form in the CEVS. It requires the user to enter their Voter ID to log into the system.



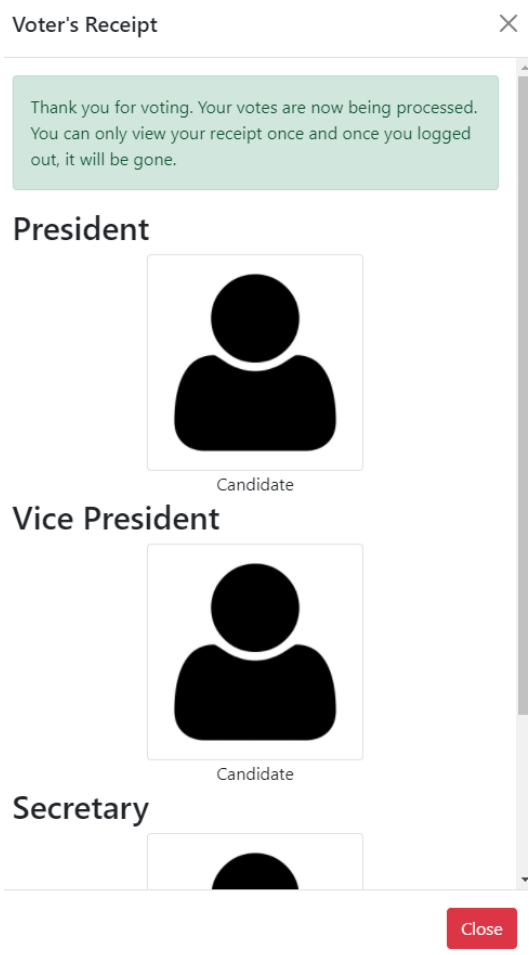
Voting Module

The Voting Module shows a ballot containing the list of the candidates with their corresponding party grouped by the position they are vying for. Voters will only be able to select one candidate for each position unless stated otherwise. The Voting Module will have the same interface for both the BEVS and CEVS.



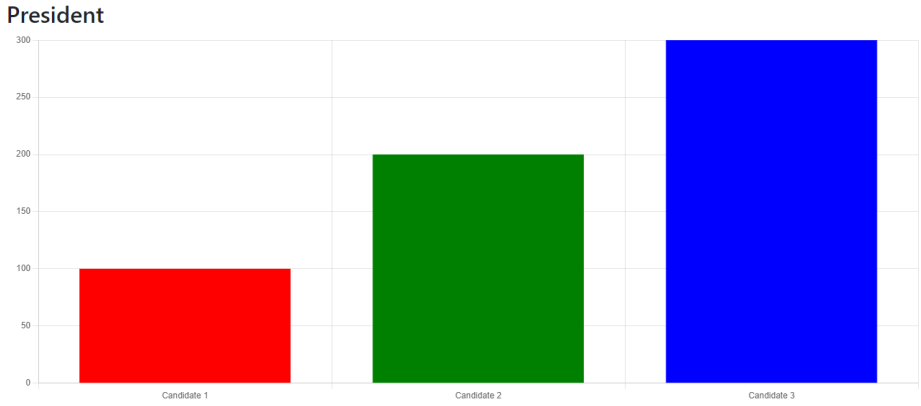
Vote Confirmation Modal

The confirmation modal allows the voter to verify his or her voting application before submitting it for processing. The system will show the list of the candidates that the voter has selected as well as a notification that upon submission of the voting application, the selected choices will be final.



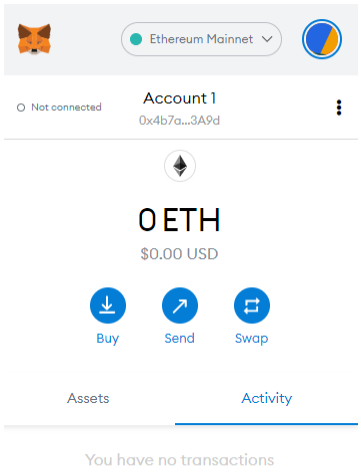
Ballot Receipt

Upon confirmation of the voter's ballot, a ballot receipt will show the candidates that the voter has casted his or her votes on. Once the modal is closed, the system will then redirect to the Results Module.



Results Module

The Results Module shows the total number of the votes of specific positions and the amount of vote percentage for each of the candidates. After a voter has submitted his or her ballot, and verified it based on their ballot receipt, a voter can already view the Results Module, which has the same interface for both the BEVS and CEVS.

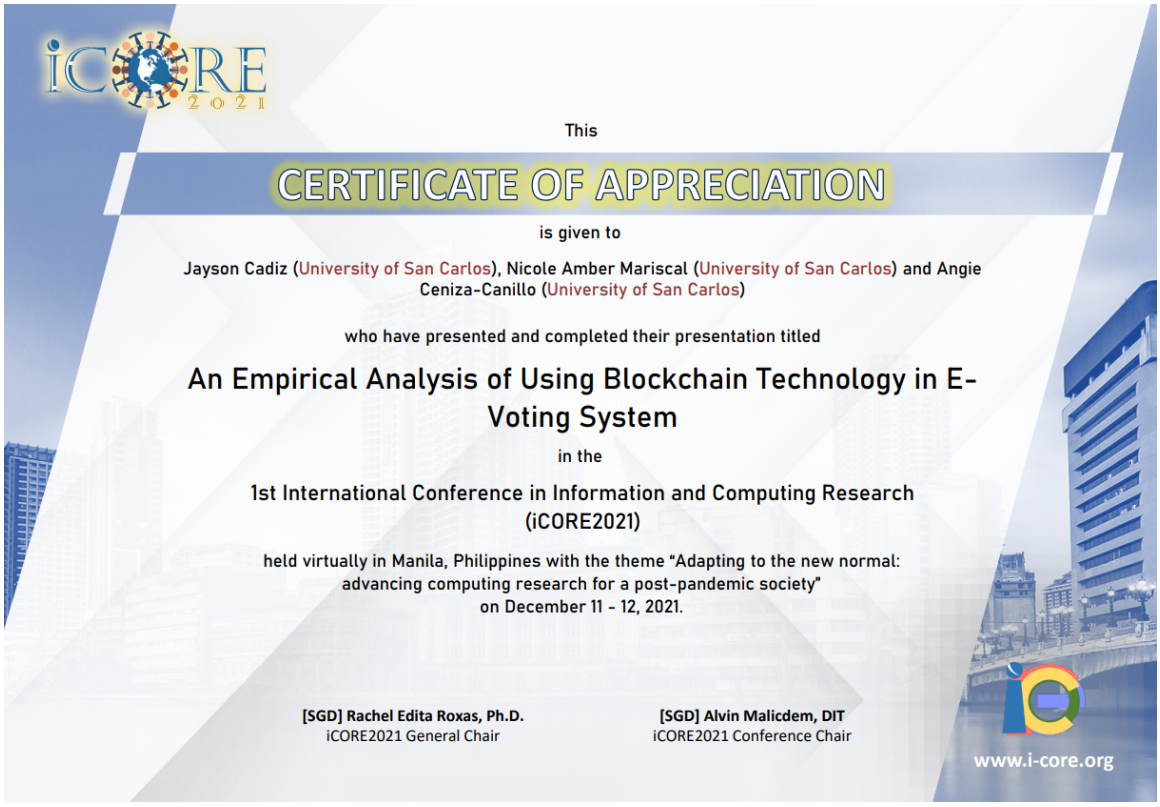


MetaMask Cryptocurrency Wallet

MetaMask is a browser extension for the cryptocurrency wallet. Metamask will be used by voters voting on the BEVS so that they can cast their votes.

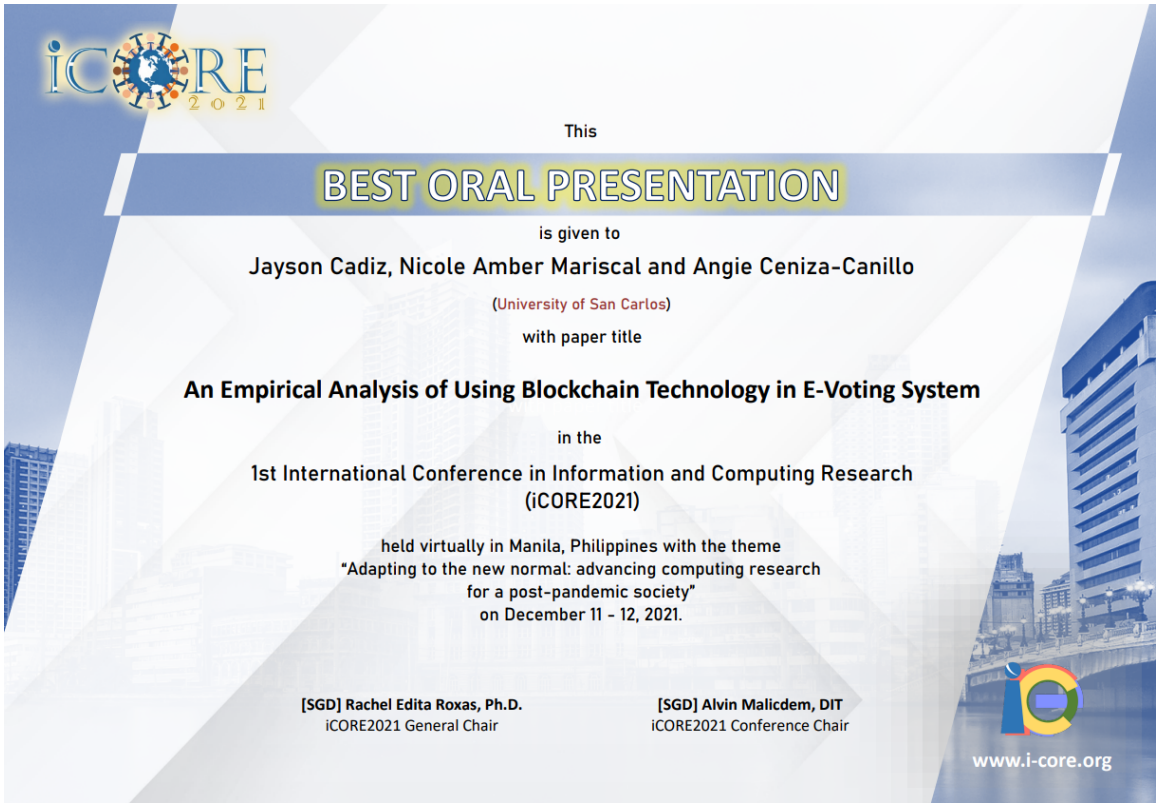
APPENDIX B

Certificates


This
CERTIFICATE OF APPRECIATION
is given to
Jayson Cadiz (University of San Carlos), Nicole Amber Mariscal (University of San Carlos) and Angie Ceniza-Canillo (University of San Carlos)
who have presented and completed their presentation titled
An Empirical Analysis of Using Blockchain Technology in E-Voting System
in the
1st International Conference in Information and Computing Research (iCORE2021)
held virtually in Manila, Philippines with the theme "Adapting to the new normal: advancing computing research for a post-pandemic society"
on December 11 - 12, 2021.
[SGD] Rachel Edita Roxas, Ph.D.
iCORE2021 General Chair
[SGD] Alvin Malicdem, DIT
iCORE2021 Conference Chair
www.i-core.org


This
CERTIFICATE OF APPRECIATION
is given to
Jayson Cadiz
(University of San Carlos)
In recognition of his/her active participation as PRESENTER for the paper titled
An Empirical Analysis of Using Blockchain Technology in E-Voting System
in the
1st International Conference in Information and Computing Research (iCORE2021)
held virtually in Manila, Philippines with the theme "Adapting to the new normal: advancing computing research for a post-pandemic society"
on December 11 - 12, 2021.
[SGD] Rachel Edita Roxas, Ph.D.
iCORE2021 General Chair
[SGD] Alvin Malicdem, DIT
iCORE2021 Conference Chair
www.i-core.org


iCORE 2021
This
CERTIFICATE OF APPRECIATION
is given to
Nicole Amber Mariscal
(University of San Carlos)
In recognition of his/her active participation as PRESENTER for the paper
titled
An Empirical Analysis of Using Blockchain Technology in E-Voting System
in the
1st International Conference in Information and Computing Research (iCORE2021)
held virtually in Manila, Philippines with the theme "Adapting to the new normal: advancing computing research for a post-pandemic society"
on December 11 - 12, 2021.
[SGD] Rachel Edita Roxas, Ph.D.
iCORE2021 General Chair
[SGD] Alvin Malicdem, DIT
iCORE2021 Conference Chair
www.i-core.org


iCORE 2021
This
BEST ORAL PRESENTATION
is given to
Jayson Cadiz, Nicole Amber Mariscal and Angie Ceniza-Canillo
(University of San Carlos)
with paper title
An Empirical Analysis of Using Blockchain Technology in E-Voting System
in the
1st International Conference in Information and Computing Research (iCORE2021)
held virtually in Manila, Philippines with the theme
"Adapting to the new normal: advancing computing research for a post-pandemic society"
on December 11 - 12, 2021.
[SGD] Rachel Edita Roxas, Ph.D.
iCORE2021 General Chair
[SGD] Alvin Malicdem, DIT
iCORE2021 Conference Chair
www.i-core.org

APPENDIX C

Published Paper/Manuscript

An Empirical Analysis Of Using Blockchain Technology In E-Voting Systems

Jayson V. Cadiz, Nicole Amber M. Mariscal, Angie M. Ceniza-Canillo
Department of Computer and Information Sciences and Mathematics
University of San Carlos
Cebu City, Philippines
16100432@usc.edu.ph, 18101346@usc.edu.ph, amceniza@usc.edu.ph

***Abstract*—An analysis on the use of blockchain technology in e-voting systems has been done to compare its performance and security to that of a centralized e-voting system. Blockchain technology is one of the emerging technologies of today as a distributed, decentralized database technology, providing much promise to transparency and reducing cybersecurity risks. A Blockchain-based E-Voting System (BEVS) and a Centralized E-Voting System (CEVS) were developed, and compared in terms of performance and security testing. The results show that the BEVS processes requests slightly slower than the CEVS. This is due to the additional processes of block validation and creation in a blockchain network, whereas utilizing a local private blockchain as the database of an e-voting system handles requests faster than utilizing a public blockchain network. However, the BEVS is more reliable when it comes to the efficiency of an e-voting system as it has a complete 0.00% error rate compared to the CEVS with its errors generated by the internal server. The speed at which the BEVS may also be affected by heavy loads, however, all requests were fulfilled, and no system crashes occurred. The BEVS also proves to be a more secure e-voting system with less vulnerabilities detected.**

Keywords-blockchain, e-voting systems, performance, cybersecurity, distributed systems

I. INTRODUCTION

Voting is a key component of a democracy. Traditionally, a paper ballot system is faced with challenges, such as the inevitable possibility of human errors that affect the credibility of an election. To eliminate that, elections in many developing countries and organizations have shifted to electronic voting or e-voting. E-voting provides ease by a quick voting process and an automatic mechanism for producing results. In the Philippines, e-voting has been implemented since the 2010 national elections. COMELEC utilized automated election systems, wherein registered voters cast their vote by shading the ovals on printed ballots and feeding their completed ballots into vote counting machines (VCMs). Unfortunately, the implementation of these machines suffered from technical and procedural difficulties. In fact, the Philippines have suffered its worst-ever government data breach barely a month before the 2016 elections, leaking personal information of over 70 million registered voters. The attack was claimed to have been made by the Anonymous Philippines in their efforts to highlight the vulnerabilities of the system, including the use of automated voting machines for the upcoming elections that year [1]. The 2019 midterm elections also suffered from technical difficulties brought about by the VCMs, ranging from paper jams to defective machines, that further delayed the voting process and even resulted in manual counting and verification of ballots [2]. As the nation continues to suffer technical issues during the elections, educational institutions implementing online classes are now also faced with the challenges of e-voting. In April 2021, the University of San Carlos - Commission on Elections (USC COMELEC) conducted its first ever online elections for the university's annual Supreme Student Council elections. USC COMELEC received reports of technical difficulties experienced by students that led to their inability to vote [3]. These included reports of student voters not receiving their one-time password (OTP) that were needed to vote, crashing of the voting site, failure of the site to produce ballot receipts and fluctuations of votes noticed by poll watchers [4]. Due to these technical issues, the USC COMELEC declared the elections a failure. A special election was conducted shortly after a month. This shows how vulnerable a centralized e-voting system is, where the main server is susceptible to cyberattacks.

A potential solution to the difficulties of e-voting systems is the integration of blockchain technology. Blockchain technology is a decentralized, distributed database technology wherein there is no central unit that takes complete control of the blockchain [5]. Given its decentralized and anonymous nature, it is nearly impossible to tamper with data that have been added in the blockchain. A blockchain-based e-voting system significantly increases the level of security

due to its validation process using the consensus algorithm, an algorithm that verifies the contents of a newly created block before adding it to the blockchain.

The objectives of this paper is to implement an e-voting system using blockchain technology and provide an empirical analysis of the blockchain-based e-voting system in comparison to a centralized e-voting system. Both a Blockchain-based E-Voting System (BEVS) and a Centralized E-Voting System (CEVS) will be developed, and analyzed by conducting tests that simulate student elections in a university in order to evaluate and compare the performance and the security of the systems. This study will be beneficial to institutions in need of a more secure voting system compared to existing implementations. This study will also be a significant contribution to the growing field of blockchain technology.

II. Review of Related Literature

A. E-Voting Systems

Electronic voting, also called e-voting, is a form of voting with the use of an electronic system. In this method, ballot casting, recording and tabulation are done automatically with the aid of computers. The main motivation behind e-voting is to promote secret voting, convenience, accuracy, verifiability and security while substantially reducing human error in the tabulation process [6]. E-voting also makes it possible to assist voters with disabilities. In the eyes of the general public, it may be difficult to fully embrace these technologies that inherently do not disclose the actual voting process. Hence, implementations of e-voting systems require extra planning and knowledge given a specific setting.

B. Blockchain Technology

Blockchain is a decentralized, distributed database technology for transaction and data management that aims to create an environment where no third party is in control of transactions and data [4]. Mutual agreement among the participants of a blockchain network is achieved by a consensus algorithm, a core component that plays a crucial role in the safety, efficiency and stability of a blockchain [7]. A blockchain consists of blocks containing data, particularly transactions, that are chained together, and if a new transaction happens on the network, a record of that transaction is added to the ledger of each user. Blockchain is a form of distributed ledger technology in which transactions are registered using a hash, which is an immutable cryptographic signature. This assumes that if a single block in a chain is modified, it is clear that the chain has been tampered with, and this makes it difficult or impossible to alter, hack, or trick the system. A blockchain can also be made flexible in terms of development and design to implement blockchain-based solutions to modern world problems. This can be achieved by creating smart contracts. A smart contract is a concept by Nick Szabo that is a self-verifying, self-executing, tamper-resistant and immutable computer program that consists of a set of rules to be run on a blockchain [8].

C. Related Work

Yi and Das [9] investigated the nature of blockchain technology for e-voting to resolve the current e-voting system limitations, to enhance integrity and transparency, and to optimize the voting process. This research entails the development of the Blockchain-based Online Voting System (BOVS) with the use of smart contracts run on an Ethereum blockchain powered by Ganache and MetaMask. Results from their questionnaires show an overall average rating of above 90% that was given by public testers, indicating their satisfaction with the system in terms of user-friendliness, maintainability, security and speed.

III. Methodology

This section presents the operations of the research procedure, from the development process and tools to the verification and testing process of the two e-voting systems. Figure 1 and 2 show a graphical conceptual framework for the vote processing of the BEVS and CEVS respectively.

Both systems were designed to have identical user interfaces. The difference between the two systems lies in the implementation of the backend development – the BEVS uses Ganache as its local blockchain network and the Ropsten Ethereum as its cloud blockchain network, while the CEVS uses MongoDB as its database system both for local and cloud. Both systems consist of the three modules, namely the Login, Voting, and Results Modules. The Login Module is implemented differently on the systems. Users in the CEVS will use their respecting voter's id in accessing the voting application, while users in the BEVS will import their digital wallet MetaMask, and login by connecting to the blockchain network. In the Voting Module, users are shown the list of candidates they can cast their vote on, and they can verify their selection with a ballot receipt. Vote processing is done after a user confirms their ballot receipt in the CEVS or issues a vote transaction in the BEVS. Users will be redirected to view the election results in the Results module.

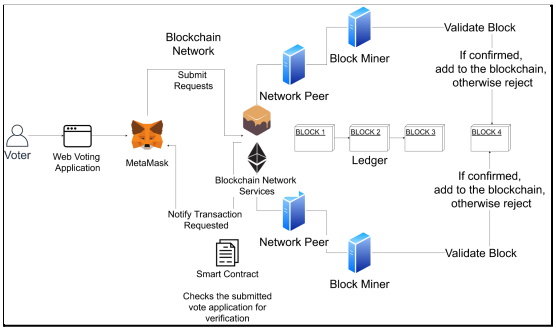


Figure 1. BEVS Conceptual Framework

In the BEVS, users in the system will use the private key of their digital wallet as their login credentials to be imported to MetaMask and connect it to the network. The MetaMask extension is required to handle the requests from the user in submitting their votes to Ganache and the Ropsten network. The smart contracts programmed in the system determine whether the request meets the terms and conditions set in the contract. If all conditions are met, a new block will be created, processed and then verified to be added to the ledger of the blockchain network that serves as the database of the system.

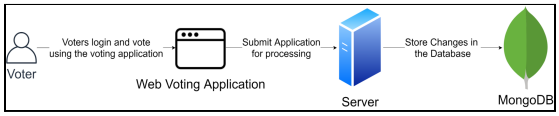


Figure 2. CEVS Conceptual Framework

In the CEVS, users are required to login using their voter ID number. Casted votes will be processed by the server, and are stored in a database that automatically tallies the vote counts of each candidate.

A. Research Procedure

Data is gathered from the testing process that measures the performance of the CEVS and the BEVS using performance testing tools, particularly Apache Jmeter with the integration of Selenium WebDriver. Both systems will be tested in two environments with respect to the nature of its database - local and cloud. The testing process is in the form of an automated simulation where first, a user sends a request to login to the system. Upon successful login, the user selects candidates, then submits their ballot for review and confirms their ballots, where a ballot receipt will also be displayed. Finally, the user is redirected to view the election results. These events are done simultaneously within a specific period of time to simulate normal to heavy loads on the system. The data is then analyzed based on performance metrics, such as the average response time and average error rate. Also included in the performance test is determining the amount of load the system can handle before becoming unresponsive.

Security testing is conducted with the use of OWASP ZAP, an open-source web application security scanner. OWASP ZAP's Automated Scan tool passively scans through the system and finds pages within a given URL. With the list of discovered pages, it proceeds to automatically attack functionalities and parameters with its active scanner. This process will generate a list of alerts with their corresponding risk levels, as well as the different instances associated with each of the alerts found in the system. Data gathered from this assessment are then analyzed and compared with the two systems.

1) Average Response Time: This is a direct measure of the average of the total amount of time the system takes to respond to a request sent by the client within a given time period.

$$ART = \frac{\sum(time\ of\ response - time\ on\ input)}{number\ of\ responses\ in\ selected\ time\ period} \quad (1)$$

2) Average Error Rate: This is a calculation that generates the average percentage of failed requests compared to the total number of requests. The percentage reflects HTTP status codes indicating an error on the server, as well as any request that times out before completing its response.

$$AER = \sum\left(\frac{failed\ events}{successful\ events + failed\ events}\right) \times 100 \quad (2)$$

3) Breaking Point: This is used to determine the maximum load-bearing capacity of the system. Several automated requests will be sent to the system, simulating many active virtual users, and the breaking point of the system will be measured as the number of requests at which the system crashes.

IV. Results and Analysis

The following tables and figures present the results from the performance test and security test of the two systems. Testing for the average response times was conducted in two different testing environments. All tests are run on a sample of 50 threads in a time period of 250 seconds, with a new thread executing every 5 seconds. For the Login Modules of both systems, the methods on performing the login process are distinct from each other. The CEVS checks for existing voter ID on the database while the BEVS only checks if MetaMask is connected to the server without prompting any requests to the blockchain network. Hence, recording the performance of the Login Module is irrelevant to the discussion.

A. Performance Testing on a Local Database

This subsection presents the performance test results of running the CEVS on a local MongoDB database and the BEVS on Ganache, a local private Ethereum blockchain. Each test done on the CEVS is composed of 50 threads, that is 50 virtual users, and in the BEVS, 50 different blockchain accounts simultaneously perform 1 transaction. These are the event processes per test. Performance data are gathered through analyzing the average processing time of submitted ballots in the Voting Module and average processing time of the retrieval of results in the Results Module, as well as the average error rates per module and breaking points of both systems. The breaking point performance test is done by constantly increasing the amount of simultaneous threads for the CEVS and transactions for the BEVS.

TABLE I. SUMMARY OF PERFORMANCE TEST RESULTS ON LOCAL VOTING MODULE.

Tests	Successful Average Response Time		Error Rate	
	BEVS	*CEVS*	*BEVS*	*CEVS*
T1	1.745 seconds	0.077 seconds	0.00%	2.00%
T2	1.683 seconds	0.071 seconds	0.00%	0.00%
T3	1.780 seconds	0.056 seconds	0.00%	0.00%
T4	1.885 seconds	0.035 seconds	0.00%	2.00%
T5	1.705 seconds	0.055 seconds	0.00%	6.00%
T6	1.822 seconds	0.289 seconds	0.00%	10.00%
T7	1.694 seconds	0.030 seconds	0.00%	2.00%
T8	1.649 seconds	0.049 seconds	0.00%	2.00%
T9	1.910 seconds	0.168 seconds	0.00%	4.00%
T10	1.684 seconds	0.213 seconds	0.00%	8.00%

Table 1 presents the summary of test results for the Voting Module of the two systems. Each test simulates 50 virtual users casting their votes in the Voting Module. The average response time recorded is the average of the time from when a user reviews their votes in the Voting Module until the time a ballot receipt is issued, signifying a successful transaction of votes. In the CEVS, the errors that occured in this module were logged with an HTTP status code of 500 - Internal Server Error. These errors were manifested by the internal server during processing of vote requests that have failed on the server side and are taken care through error handling with the aim to enable the application to proceed despite manifesting errors. In the event of these errors, the casted votes of that account were not counted, thus bearing no change to the election results. Although the CEVS processes requests faster as shown in Table 2, the BEVS shows promise with its consistent 0.00% error rate. The BEVS in its local environment, processes vote requests at a slightly slower response time. Although block validation is done almost instantly, the time it takes to create and process transactions is the source of these response times while still performing the actual processes as expected in a blockchain network.

TABLE II. SUMMARY OF PERFORMANCE TEST RESULTS ON LOCAL RESULTS MODULE

Tests	Successful Average Response Time		Error Rate	
	BEVS	*CEVS*	*BEVS*	*CEVS*
T1	0.609 seconds	0.644 seconds	0.00%	0.00%
T2	0.638 seconds	0.885 seconds	0.00%	0.00%
T3	0.831 seconds	0.841 seconds	0.00%	0.00%
T4	0.650 seconds	0.690 seconds	0.00%	0.00%
T5	0.618 seconds	0.760 seconds	0.00%	0.00%
T6	0.645 seconds	0.923 seconds	0.00%	0.00%
T7	0.591 seconds	0.619 seconds	0.00%	0.00%
T8	0.598 seconds	0.643 seconds	0.00%	0.00%
T9	0.660 seconds	1.034 seconds	0.00%	0.00%
T10	0.544 seconds	1.510 seconds	0.00%	0.00%

The average response time monitored in Table 2 is the time from when the server completes vote processing to the time the updated results are shown to the user in the browser. Although the page instantly loads and displays page content, the actual count of votes takes a while to appear on the charts. This highlights how long it takes to retrieve data from the server. There is only a slight difference in the average times of both systems. Although the CEVS had the fastest average time of 0.168 seconds in T9, internal server errors occurred in 4.00% of the total number of threads executed. In the event of these errors, the Results Module was displayed with an empty chart. This is significantly comparable to the BEVS with no errors despite its slower response time.

B. Performance Testing on Cloud Database

This subsection presents the performance test results of running the CEVS on a cloud database using MongoDB Atlas and the BEVS on Ropsten, a public blockchain test network. Testing procedures are done similarly with the local environment.

TABLE III. SUMMARY OF PERFORMANCE TEST RESULTS ON CLOUD VOTING MODULE

Tests	Successful Average Response Time		Error Rate	
	BEVS	*CEVS*	*BEVS*	*CEVS*
T1	50.7 seconds	1.683 seconds	0.00%	16.00%
T2	19 seconds	1.701 seconds	0.00%	24.00%
T3	17.9 seconds	1.862 seconds	0.00%	14.00%
T4	28.7 seconds	1.908 seconds	0.00%	22.00%
T5	17.3 seconds	1.681 seconds	0.00%	14.00%
T6	21.2 seconds	1.727 seconds	0.00%	8.00%
T7	17.7 seconds	1.943 seconds	0.00%	12.00%
T8	17.9 seconds	1.822 seconds	0.00%	18.00%
T9	15.4 seconds	1.754 seconds	0.00%	16.00%
T10	19.4 seconds	1.706 seconds	0.00%	20.00%

The summary of test results for the systems' Voting Module on a cloud database are shown in Table 3. In the CEVS, frequent errors were encountered in this module. Errors were logged with

an HTTP status code of 500, similar to the errors that occured in the tests performed on the CEVS with a local database. The casted votes of erroneous threads were not counted, thus bearing no change to the election results. On the console terminal of the text editor from which the server was being run, an error message appeared indicating a Mongo Server error. Upon further investigation, the errors were caused by a limitation in MongoDB Atlas which caters to only 500 concurrent incoming connections per node. Any subsequent requests that require database access were inoperative. A server restart via console terminal was required and frequently applied to resume each test. On the other hand, the BEVS test results show the time from which the system processes vote transactions into a block to the time when the block is validated and returns a successful block transaction. Similar to the performance test conducted on a local database and blockchain network, the CEVS on a cloud database processes votes faster. The BEVS running on a public test network performs extremely slow compared to its local counterpart as transactions in the public blockchain network are processed in a block together with other transactions in the network before being validated.

TABLE IV. SUMMARY OF PERFORMANCE TEST RESULTS ON CLOUD RESULTS MODULE

Tests	Successful Average Response Time		Error Rate	
	BEVS	*CEVS*	*BEVS*	*CEVS*
T1	1.495 seconds	1.270 seconds	0.00%	12.00%
T2	1.466 seconds	1.257 seconds	0.00%	12.00%
T3	1.456 seconds	1.472 seconds	0.00%	10.00%
T4	1.404 seconds	1.362 seconds	0.00%	12.00%
T5	1.376 seconds	1.376 seconds	0.00%	2.00%
T6	1.356 seconds	1.296 seconds	0.00%	4.00%
T7	1.387 seconds	1.292 seconds	0.00%	10.00%
T8	1.377 seconds	1.369 seconds	0.00%	4.00%
T9	1.367 seconds	1.243 seconds	0.00%	8.00%
T10	1.413 seconds	1.598 seconds	0.00%	12.00%

The average response time monitored in Table 4 is the time from when the server completes vote processing to the time the updated results are shown to the user in the browser. Similar to the tests in the local environment, there is only a slight difference in the average times of both systems, with the CEVS having the fastest average time. However, the CEVS has constant server errors as compared to the BEVS with no errors despite a slightly slower response time. The Results Module did not display updated results as expected in the event of such errors.

C. *Average Error Rate and Breaking Point*

This subsection highlights the average error rates in each process of the two systems. The CEVS in both the local database and cloud database have a higher percentage of average error rates compared to that of the BEVS as shown in Table 5. This proves that the CEVS is more prone to server errors, specifically in the Voting Module, that is a core component in an e-voting system.

TABLE V. SUMMARY OF AVERAGE ERROR RATES

Test	Local Database		Cloud Database	
	BEVS	*CEVS*	*BEVS*	*CEVS*
Login Module	0.00%	0.00%	0.00%	7.4%
Voting Module	0.00%	3.6%	0.00%	16.4%
Results Module	0.00%	0.00%	0.00%	8.6%

Upon further observation of both systems for their breaking points, in the local testing environment, the CEVS shows no proof of system crashes that can cause the system to become unresponsive. Multiple error occurrences were observed in the testing process. However, the observed errors did not pose any vital issues that could disrupt other requests given to the system. The system handled any error occurrences thus allowing the system to continue to perform other processes and prevent the system from becoming completely unresponsive. The BEVS as well exhibits no manifestations of the system crashing. The system barely processes requests due to high system load but nonetheless, all requests were fulfilled successfully and performance response times and request processes were affected without becoming unresponsive.

In the remote testing environment, the CEVS shows a point wherein requests were no longer processed due to constraints pertained in the server as the limit for the number of connections to the server is limited to 500 connections and thus the server can only handle the given amount of connections before the system becomes unresponsive. The BEVS on the other hand can accommodate as many requests as possible. The only occurrence that the system rejects requests is when a single account is prompting too many requests at the same time and with this occurrence, the system can still process other requests without the system crashing.

D. *Security Testing*

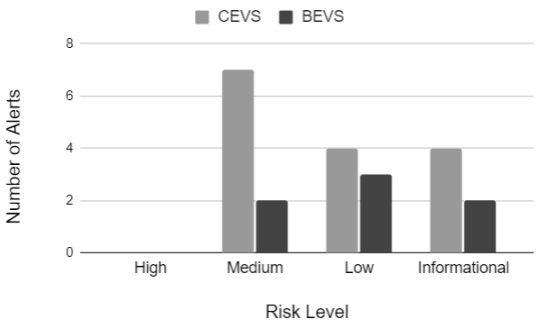


Figure 3. Summary of Alerts and Risk Levels

The security of the two systems were tested with the use of OWASP ZAP's Automated Scan tool. The number of alerts with their corresponding risk levels found on the two systems are shown in Fig 3. There is a higher number of alerts found in the CEVS, with most alerts of a medium risk level. Note that some alerts found in the CEVS are listed twice, as these alerts were detected on both the server and the frontend application during the scanning procedure.

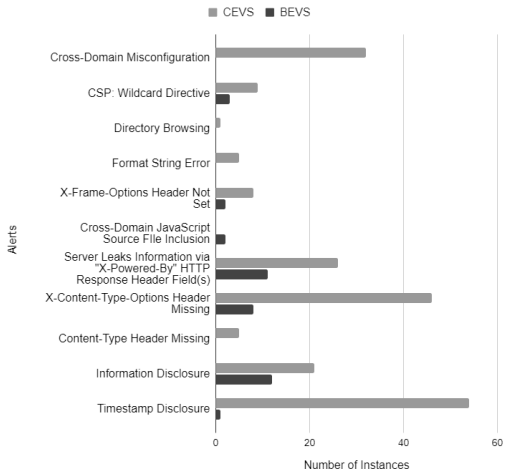


Figure 4. Summary of Alerts and Instances from Security Scan Report

Different instances of the alerts, arranged by risk level from Medium to Informational, are shown in Fig. 4. There are more vulnerabilities found in the CEVS compared to the BEVS. Although most of the alerts were detected in both systems, there is a significantly higher number of instances found in the CEVS.

Upon further research into the alerts generated by the security scan, the risks of each alert are listed in Table 12.

TABLE VI. SUMMARY OF ALERTS AND RISKS

Alerts	Risks
Cross-Domain Misconfiguration	CORS-based attacks
CSP: Wildcard Directive	Cross-site Scripting (XXS) and data injection attacks
Directory Browsing	Brute force attacks
Format String Error	Format string exploits
X-Frame-Options Header Not Set	Clickjacking attacks
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Exploits on frameworks and components of a web application
X-Content-Type-Options Header Missing	Cross-site Scripting (XXS) attacks
Content-Type Header Missing	Cross-site Scripting (XXS) attacks
Information Disclosure	Suspicious comments that may help an attacker understand underlying application logic and possible weakness, thus allowing an attacker to create a malicious attack
Timestamp Disclosure	Sensitive information retrieval

V. CONCLUSION AND RECOMMENDATION

An analysis has been made on the implementation of blockchain technology on e-voting systems. This is done by developing a centralized e-voting system and a blockchain-based e-voting system, and testing both systems to scale and compare their performance and security. The concluding observations from this research are listed below.

- The performance of the BEVS is better with the use of a local blockchain network than that of a public blockchain network.
- The performance of the BEVS is slower by at least 1 second compared to the CEVS. However, the BEVS is more reliable given its lower error rates.
- The BEVS is more secure given the few vulnerabilities that were found in the security scan as compared to the CEVS.

Future researchers should conduct testing in a live controlled group for a more practical and realistic approach. Future studies can also implement e-voting systems for a bigger scale of users, possibly simulating a nationwide voting scenario, and investigate its scalability. Security evaluations could also be done in-depth, such as specifically targeting the OSI layers of the network. Another recommendation is to make use of other consensus algorithms and blockchain types that best fit the requirements of the system.

ACKNOWLEDGMENT

The researchers thank the Department of Computer and Information Sciences and Mathematics of the School of Arts and Sciences in the University of San Carlos for allowing us to conduct this research, and for guiding us in the revision of the concept of this study.

REFERENCES

[1] L. Chi, "Philippine elections hack 'leaks voter data,'" *BBC News*, April 2016. [Online]. Available: https://www.bbc.com/news/technology-36013713

[2] M. Espina, "Election 2019 technical issues: Paper jams, malfunctioning machines." *Rappler*, May 2019. [Online]. Available: https://www.rappler.com/nation/elections/vote-counting-machines-errors-technical-issues

[3] USC COMELEC Investigation Committee, "COMELEC investigation committee report on the failure of the 2021 USC-SSC elections," May 2021. [Online] Available: https://www.facebook.com/usccomelec/posts/4013553535349177

[4] P. Webster, "First as tragedy, then as farce: what happened to the 2021 USC-SSC elections?," May 2021 [Online] Available:

https://www.facebook.com/todayscarolinian/posts/10159360219518814

[5] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, “Where is current research on blockchain technology? - a systematic review,” *PLoS ONE*, vol. 11, pp. 1-27, October 2016. Available: doi: 10.1371/journal.pone.0163477

[6] E. Abu-Shanab, M. Knight, and H. Refai, “E-voting systems: a tool for e-democracy,” *Management Research and Practice*, vol. 2, pp. 264 - 274, 2010. Available: https://www.researchgate.net

[7] D. Mingxiao, M. Xiaofeng, Z. Zhe. W. Xiangwei, and C. Qijun, “A review on consensus algorithm of blockchain,” *IEEE SMC*, pp. 2567-2572, 2017. Available: doi: 10.1109/SMC.2017.8123011

[8] B. Mohanta, S. Panda, and D. Jena, “An overview of smart contract and use cases in blockchain technology,” *ICCCNT*, 2018. Available: doi: 10.1109/ICCCNT.2018.8494045

[9] O. K. Yi, and D. Das, "Blockchain technology for electronic voting," *Journal of Critical Reviews*, vol. 7, pp. 114-124, 2020. Available doi: 10.31838/jcr.07.03.22

CURRICULUM VITAE

CONTACT INFORMATION

Name: Jayson Cadiz
Address: Blk 14 Lot 25 La Aldea Buena Mactan
Lapu-Lapu City
Telephone: 268-0391
Cellphone: 09324152157
Email: cadizjayson25@yahoo.com

PERSONAL INFORMATION

Birthday: June 2, 2000
Religion: Roman Catholic
Civil Status: Single

EDUCATION

University of San Carlos Talamban Campus
Bachelor of Science in Computer Science
Tertiary Level (2016 - present)

University of San Carlos North Campus
Secondary Level (2012 - 2016)

University of San Carlos North Campus
Primary Level (2010 - 2012)

Marie Ernestine School
Primary Level (2005 - 2010)

STRENGTHS/TRAITS & SKILLS

- High degree of initiative
- Hands-on experience
- Good problem analysis skills
- Identify problems and provide feasible solutions
- Good interpersonal skills
- Able to work within tight schedules
- Good leadership skills

TECHNICAL SKILLS

- NodeJs
- Python
- HTML & CSS
- PHP
- MySQL
- JavaScript
- C
- Vue

CURRICULUM VITAE

CONTACT INFORMATION

Name: Nicole Amber M. Mariscal

Address: Blk 2, Lot 4 Sola Dos Subdivision,

Talamban, Cebu City, Cebu, Philippines

Telephone: 232-5735

Cellphone: 09760567437

Email: mnicoleamber21@gmail.com

PERSONAL INFORMATION

Birthday: July 21, 1999

Religion: Roman Catholic

Civil Status: Single

EDUCATION

University of San Carlos

Bachelor of Science in Computer Science

Tertiary Level (2018 - present)

Colegio de la Inmaculada Concepcion-Mandaue

Secondary Level (2012 - 2018)

Colegio de la Inmaculada Concepcion-Mandaue

Primary Level (2006 - 2012)

ACHIEVEMENTS

2018 - 2020 Dean’s List

2018 With High Honors

2018 Journalism Award

2016 2nd runner up in Feature Writing in Filipino in Regional Schools Press Conference (RSPC)

STRENGTHS/TRAITS & SKILLS

- High degree of initiative
- Good communication skills
- Strong presentation skills
- Good leadership skills
- Cooperative team player

TECHNICAL SKILLS

- HTML & CSS
- PHP
- C
- Express.js
- ReactJS
- MySQL
- Python
- Flutter
- Git