A Novel Electronic Voting System Based on Block Chain

A Sharmila [1], B Mahesh [2]
[1] Dept of Computer Science And Engineering
[2]Asst. Professor, Dept of Computer Science And Engineering
[1, 2] Seshachala Institute Of Technology, Puttur, A.P, India

Abstract- *Building a secure electronic voting system that offers the fairness and privacy of current voting schemes, while providing the transparency and flexibility offered by electronic systems has been a challenge for a long time. In this work-in-progress paper, we evaluate an application of block chain as a service to implement distributed electronic voting systems. The paper proposes a novel electronic voting system based on block chain that addresses some of the limitations in existing systems and evaluates some of the popular block chain frameworks for the purpose of constructing a block chain-based e-voting system. In particular, we evaluate the potential of distributed ledger technologies through the description of a case study; namely, the process of an election, and the implementation of a block chain based application, which improves the security and decreases the cost of hosting a nationwide election.*

I. INTRODUCTION

E-voting is among the key public sectors that can be disrupted by block chain technology. The idea in block chain-enabled e-voting (BEV) is simple. To use a digital-currency analogy, BEV issues each voter a "wallet" containing a user credential. Each voter gets a single "coin" representing one opportunity to vote. Casting a vote transfers the voter's coin to a candidate's wallet. A voter can spend his or her coin only once. However, voters can change their vote before a preset deadline. Here, we argue that block chains might address two of the most prevalent concerns in voting today: voter access and voter fraud. The idea is as follows. Eligible voters cast a ballot anonymously using a computer or smart phone. BEV employs an encrypted key and tamperproof personal IDs. For example, the mobile e-voting platform of the Boston-based startup Voatz employs smart biometrics and real-time ID verification. The public ledger ties each cast ballot to an individual voter and establishes a permanent, immutable record. No bad actor can engage in nefarious activities because such activities will be evident on the ledger or corrected by a peer-to-peer consensus network. To compromise the network, hackers would need to successfully hack most of the blocks (files with transaction records) before new blocks were introduced. The block chain's audit trail ensures that no vote has been changed or removed and that no fraudulent and illegitimate votes have been added. Put simply, block chains enable the creation of tamper-proof audit trails for voting. In this article, we highlight some BEV implementations and the approach's potential benefits and challenges.

1.1. Existing System

Electronic voting (also known as e-voting) refers to voting using electronic means to either aid or take care of the chores of casting and counting votes.

Depending on the particular implementation, e-voting may use standalone electronic voting machines (also called EVM) or computers connected to the Internet. It may encompass a range of Internet services, from basic transmission of tabulated results to full-function online voting through common connectable household devices. The degree of automation may be limited to marking a paper ballot, or may be a comprehensive system of vote input, vote recording, data encryption and transmission to servers, and consolidation and tabulation of election results.

A worthy e-voting system must perform most of these tasks while complying with a set of standards established by regulatory bodies, and must also be capable to deal successfully with strong requirements associated with security, accuracy, integrity, swiftness, privacy, audit ability, accessibility, cost-effectiveness, scalability and ecological sustainability.

The vast majority of the ongoing work discusses security, exactness, respectability, quickness, protection, and review capacity however existing frameworks are powerless for assaults at some degree.

Disadvantages of Existing System

- Centralized architecture.
- Attack prone.
- Not trustable.

- Non-transparent vote casting process.

1.2. Problem Statement

The present technique requires an aggressor connect specifically with the casting a ballot procedure to disturb it. On the other end, Internet is harder to control and deal with the security as Network and web related assaults are harder to follow.

1.3. Proposed system

Election Polling is a complex system as well as costly system. Here we are presenting a novel Secure, Privacy Preserving and cost effective election polling concept which uses Web Technology with GPRS Connectivity, Cloud Data Storage and Homomorpic encryption.

This system has two types of users one is Election Officer & another is Booth Manager, Booth Manager system developed with voters functionality where voters are going to poll.

Election officer will act as an admin user and he has to do the setting and configuration setting for election polling. Booth Managers are the area manages those who are responsible to add the voters details into the system and has retrieval system by which they can able to view the voted candidate details and sum of the votes.

Voters has to go the Booth where the Booth manager verify the voter and allow him to poll on the Booth's Laptop where the our voting system is running.

This proposed system has a method to execute operations on encrypted data without decrypting them which will provide us with the same results after calculations as if we have worked directly on the raw data.

Advantage of Proposed System

- Decentralized architecture.
- Transparent vote casting process.
- Manipulations of votes are nearly impossible.
- Votes are recorded accurately, permanently, securely, and transparently.

II. LITERATURE SURVEY

1: Technology-Assisted Review Makes Main Street

In our last article, we wrote about various analytics tools available in software platforms for legal document review. One tool we identified is technology-assisted review (also known as "TAR" or "predictive coding"). Six or seven years ago litigants began using TAR with little assurance that courts would accept the methodology. Since then, numerous U.S. and international courts have accepted the use of TAR. In short, TAR is now considered mainstream.

2: Random Oracles are Practical: A Paradigm for Designing E_cient Protocols

We argue that the random oracle model |where all parties have access to a public random oracle| provides a bridge between cryptographic theory and cryptographic practice. In the paradigm we suggest, a practical protocol P is produced by _rst devising and proving correct a protocol PR for the random oracle model, and then replacing oracle accesses by the computation of an \appropriately chosen" function h. This paradigm yields protocols much more e_cient than standard ones while retaining many of the advantages of provable security. We illustrate these gains for problems including encryption, signatures, and zero-knowledge proofs.

3: A simple unpredictable pseudo-random number generator

Two closely-related pseudo-random sequence generators are presented: The lIP generator, with input P a prime, outputs the quotient digits obtained on dividing by P. The x mod Ngenerator with inputs N, Xo (where N P. Q is a product of distinct primes, each congruent to 3 mod 4, and x0 is a quadratic residue mod N), outputs bob1 b2" where bi parity (xi) and xi+ x mod N. From short seeds each generator efficiently produces long well-distributed sequences. Moreover, both generators have computationally hard problems at their core. The first generator's sequences, however, are completely predictable (from any small segment of 21PI + consecutive digits one can infer the "seed," P, and continue the sequence backwards and forwards), whereas the second, under a certain intractability assumption, is unpredictable in a precise sense. The second generator has additional interesting properties: from knowledge of Xo and N but not P or Q, one can generate the sequence forwards, but, under theabove-mentioned intractability assumption, one can not generate the sequence backwards. From the additional knowledge of P and Q, one can generate the sequence backwards; one can even "jump" about from any point in the sequence to any other. Because of these properties, the x mod Ngenerator promises many interesting applications, e.g., to public-key cryptography. To use these generators in practice, an analysis is needed of various properties of these sequences such as their periods. This analysis is begun here.

4:A fully homomorphic encryption scheme

We propose a solution to the old open problem of constructing a fully homomorphic encryption scheme. This notion, originally called a privacy homomorphism, was introduced by Rivest, Adleman and Dertouzous shortly after the invention of RSA by Rivest, Shamir, and Adleman [121]. Basic RSA is a multiplicatively homomorphic encryption scheme – i.e., given RSA public key pk = (N, e) and ciphertexts {ψi ← π e i mod N}, one can efficiently compute Q i ψi = (Q i πi) e mod N, a ciphertext that encrypts the product of the original plaintexts. One imagines that it was RSA's multiplicative homomorphism, an accidental but useful property, that led Rivest et al. [120] to ask a natural question: What can one do with an encryption scheme that is fully homomorphic: a scheme E with an efficient algorithm EvaluateE that, for any valid public key pk, any circuit C (not just a circuit consisting of multiplication gates as in RSA), and any ciphertexts ψi ← EncryptE (pk, πi), outputs ψ ← EvaluateE (pk, C, ψ1, . . . , ψt) , a valid encryption of C(π1, . . . , πt) under pk? Their answer: one can arbitrarily compute on encrypted data – i.e., one can process encrypted data (query it, write into it, do anything to it that can be efficiently expressed as a circuit) without the decryption key. As an application, they suggested private data banks. A user can store its data on an untrusted server in encrypted form. Later, it can send a query on the data to the server, whereupon the server can express this query as a circuit to be applied to the data, and use the EvaluateE algorithm to construct an encrypted response to the user's query, which the user then decrypts. We obviously want the server's response here to be more concise than the trivial solution, in which the server just sends all of the encrypted data back to the user to process on its own. Cryptographers have accumulated a long assortment of "killer" applications for fully homomorphic encryption since then. However, until now, we did not have a viable construction.

5: Trust and Privacy Challenges in Securing Cloud Computing Environment's

Cloud computing is known as the newest technologies in IT field which causes some worries for consumers and its producers due to its novelty. Looking at its literature, we can see the privacy and security aspects and trust are the main concerns. It creates an important hindrance for using by users. So we decided to evaluate some factors such as security for the acceptance of cloud computing. In this paper, we highlighted envision about security emphasizing for the maintenance of privacy and trust in accepting the cloud computing. As a result, we are proposed new recommendations for improving security, decreasing risks, increasing trust and maintaining privacy which they are necessary to adopt cloud computing.

III. SYSTEM ARCHITECTURE AND DESIGN

3.1. System design

System Architecture design-identifies the overall hypermedia structure for the WebApp. Architecture design is tied to the goals establish for a WebApp, the content to be presented, the users who will visit, and the navigation philosophy that has been established. Content architecture, focuses on the manner in which content objects and structured for presentation and navigation. WebApp architecture, addresses the manner in which the application is structure to manage user interaction, handle internal processing tasks, effect navigation, and present content. WebApp architecture is defined within the context of the development environment in which the application is to be implemented.

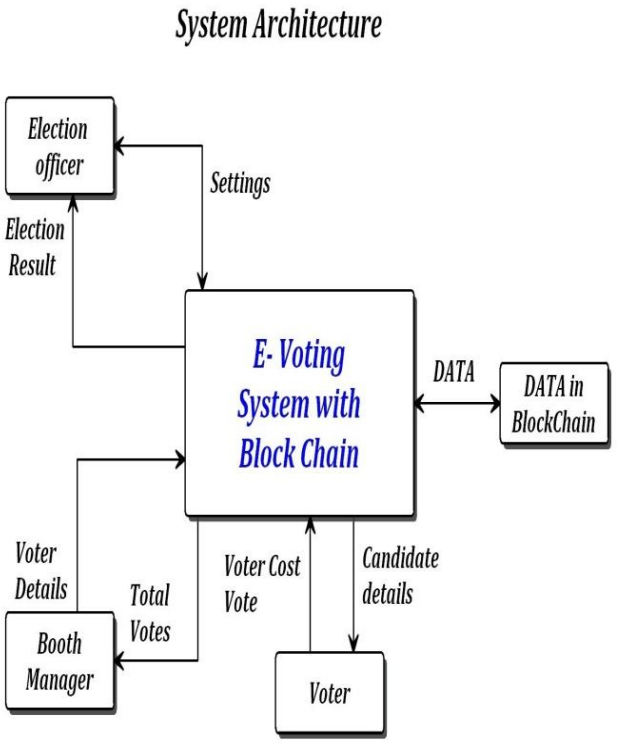


Fig 3.1 System Architecture

3.2. Use case diagrams

A use case is a set of scenarios that describing an interaction between a source and a destination. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors. shows the use case diagram.

3.2.1. Use case diagram Admin

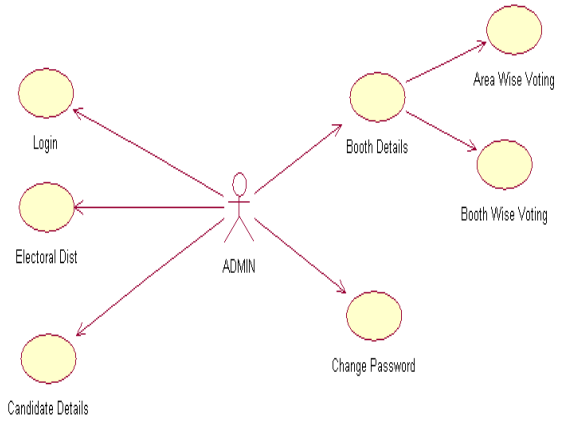


Fig 3.2.1 Use Case Diagram Admin

3.2.2. Use case diagram Booth Manager

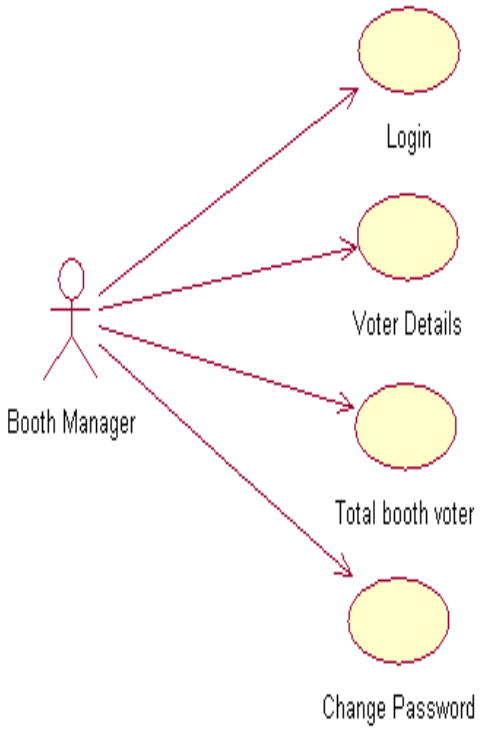


Fig 3.2.2 Use Case Diagram Booth Manager

3.2.3. Use case diagram voter

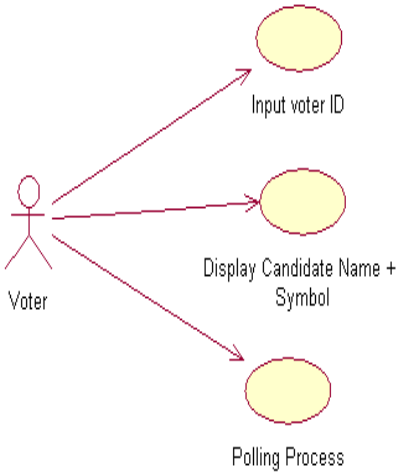


Fig 3.2.3 Use Case Diagram voter

3.3. Data flow diagram

1. A data flow diagram (DFD) is graphic representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context level DFD first which shows the interaction between the system and outside entities. DFD's show the flow of data from external entities into the system, how the data moves from one process to another, as well as its logical storage. There are only four symbols:
2. Squares representing external entities, which are sources and destinations of information entering and leaving the system.
3. Rounded rectangles representing processes, in other methodologies, may be called 'Activities', 'Actions', 'Procedures', 'Subsystems' etc. which take data as input, do processing to it, and output it.
4. Arrows representing the data flows, which can either, be electronic data or physical items. It is impossible for data to flow from data store to data store except via a process, and external entities are not allowed to access data stores directly.
5. The flat three-sided rectangle is representing data stores should both receive information for storing and provide it for further processing.

3.3.1. Level 0 data flow diagram

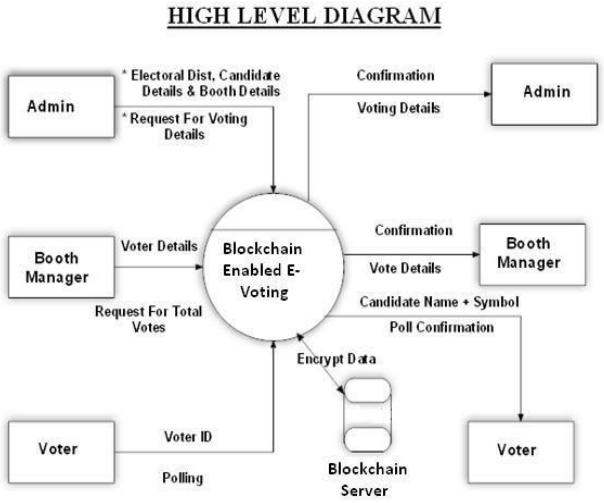


Fig 3.3.1 Level 0Data Flow diagram

3.3.2. Level1 data flow diagram

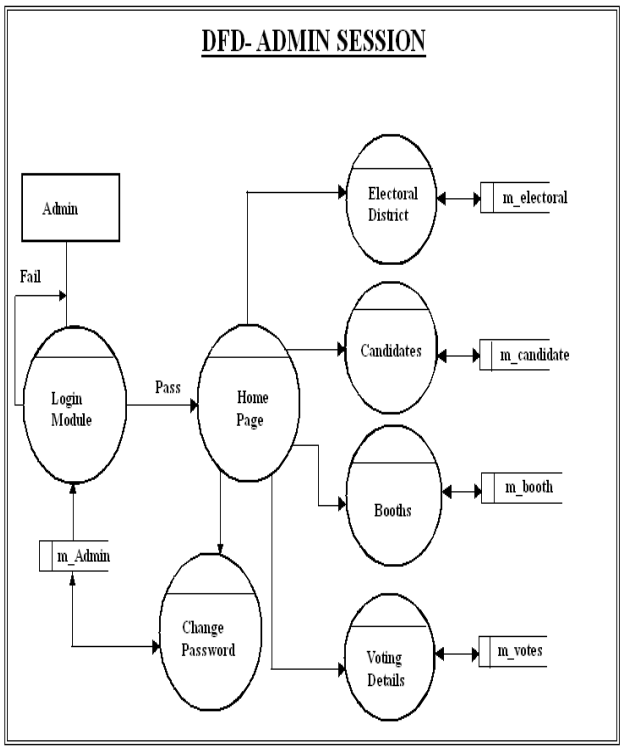


Fig 3.3.2 Level 1 Data Flow Diagram

3.3.3. Level2 data flow diagram

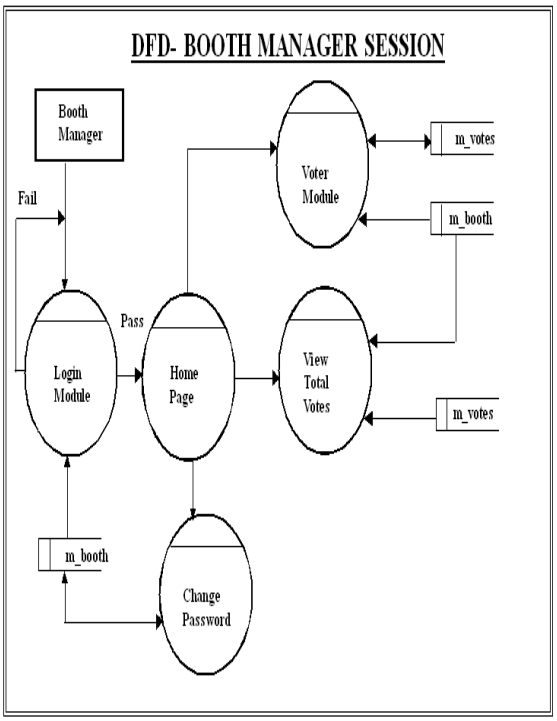


Fig 3.3.3 Level2 Data Flow Diagram

3.3.4 Level2 data flow diagram

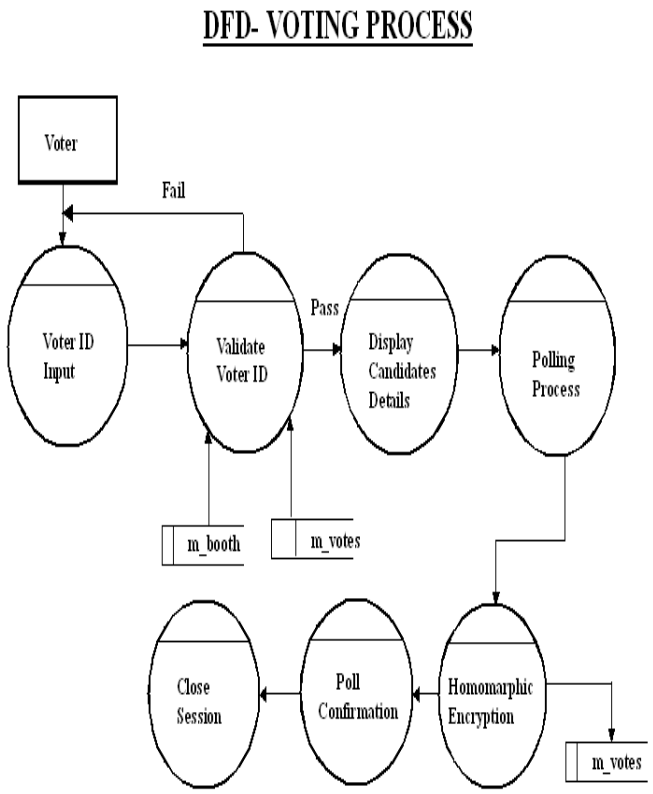


Fig 3.3.4 Level2 Data Flow Diagram

3.3.5 Level2 data flow diagram

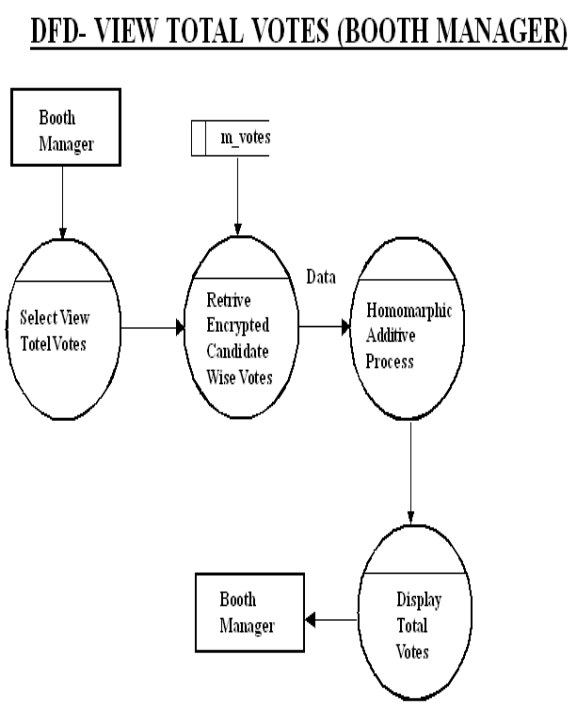


Fig 3.3.5 Level2 Data Flow Diagram

3.4.Modules Description

Modules

Electoral dist Maintenance

- Election officer has the authority to add, delete or edit the election district list. Candidate details like name, age, party, district can checked, edited, added or deleted. Likewise even the booth details like the reference number, district and the booth manager in-charge can be seen or edited. Mainly the election officer has the authority and the secret key to decrypt the individual votes of each candidate from different booth and announce the winner of election district wise.

Booth Maintenance

- Booth manager will have information about his booth regarding booth reference number, booth location, number of candidates contesting for election and total number of voters destined to vote in his booth. He has the authority to see the voter details who belong to his booth. He can add or delete any voter from the list. Voter is allowed to vote provided his voter-id is valid and cast his vote. This happens under the booth manager assistance. After voting, Booth manager can view the total number of votes, indirectly representing the total number of voters polled but individual votes per candidate can be viewed in the encrypted format.

Voter Details

- Voter details have to display as per the booth.

Voting Process

- In this module the process of voting is carried out. The voter's identity is to be validated, whether he belongs to his assigned booth and whether is has polled or not. Provided he hasn't already voted, he can cast his vote. This vote will be encrypted and added to the particular candidate to whom he/she has voted and this data is stored.

Homomorphic Encryption

- Homomorphic encryption on data, a small module is developed which shows addition, subtraction and multiplication operations on encrypted data which uses the RNS(Residue Number System) algorithm . It is easier when compared to paillier and also it is robust and more efficient.

Block chain Storage

- A block chain, originally block chain, is a growing list of records, called blocks, which are linked using cryptography.

Election Request in graph

- As per the election, result has to display in the graph.

V. TESTING AND IMPLEMENTATION

5.1 Software Testing Introduction

Software testing is a process used to help identify the correctness, completeness and quality of developed computer software. Software testing is the process used to measure the quality of developed software .Testing is the process of executing a program with the intent of finding errors. Software testing is often referred to as verification & validation

5.2 Explanation for SDLC & STLC

SDLC: The software development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

5.3 Phases of Software Development

- Requirement Analysis
- Software design
- Development or Coding
- Testing
- Maintenance

5.3.1 Requirement Analysis

The requirements of a desired software product are extracted. Based the business scenario the SRS (Software Requirement Specification) document is prepared in this phase.

5.3.2. Design

Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications, and security issues for the software. Design phase is concerned with making sure the software system will meet the requirements of the product.

There are 2 stages in design,

HLD – High Level Design
LLD – Low Level Design

HLD – gives the architecture of the software product to be developed and is done by architects and senior developers.

LLD – done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code.

5.3.3. Testing

Testing is evaluating the software to check for the user requirements. Here the software is evaluated with intent of finding defects.

5.3.4. Maintenance

Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up-to-date concerning the latest modifications and procedures.

5.4. SDLC Models

5.4.1. Water Fall Model

It will be executing one by one of the SDLC process. The design Starts after completing the requirements analysis coding begins after design. It is a traditional model It is a sequential design process, often used in SDLC, in which the progress is seen as flowing steadily downwards (like a waterfall), through the different phases.

5.4.2.Proto Type Model

Developed from the sample after getting good feed back from the customer. This is the Valuable mechanism for gaining better understanding of the customer needs.

5.4.3 Rapid Application Development Model(RAD)

This mechanism will develop from already existing one . If The New requirement is matching in already existing requirement , will develop from that**.**

5.4.4. Spiral Model

This mechanism is update the application version by version. All the SDLC process will update version by version**.**

5.4.5.V-Model

V model is a process where the development and testing phases can do parallely. For every development phase there is a testing phase. Development phases are called as verification whereas testing phases are called as validation.

5.4.6. STLC (Software Testing Life Cycle)

Testing itself has many phases i.e. is called as STLC. STLC is part of SDLC

Test	Plan
Test	Development
Test	Execution
Analyze	Results
Defect	Tracking
Summaries Report	

5.5.Test Plan

It is a document which describes the testing environment, purpose, scope, objectives, test strategy, schedules, mile stones, testing tool, roles and responsibilities, risks, training, staffing and who is going to test the application, what type of tests should be performed and how it will track the defects.

5.5.1.Test Development

Preparing test cases, test data, Preparing test procedure, Preparing test scenario, Writing test script

5.5.2.Test Execution

In this phase we execute the documents those are prepared in test development phase.

5.5.3.Analyze Result

Once executed documents will get results either pass or fail. we need to analyze the results during this phase.

5.5.4.Defect Tracking

Whenever we get defect on the application we need to prepare the bug report file and forwards to Test Team Lead and Dev Team. The Dev Team will fix the bug. Again we have to test the application. This cycle repeats till we get the software without defects.

5.6.Types Of Testing

White Box Testing
Black Box Testing
Grey box testing

5.6.1. White Box Testing

White box testing as the name suggests gives the internal view of the software. This type of testing is also known as structural testing or glass box testing as well, as the interest lies in what lies inside the box.

5.6.2.Black Box Testing

Its also called as behavioral testing. It focuses on the functional requirements of the software. Testing either functional or non functional without reference to the internal structure of the component or system is called black box testing.

5.6.3.Grey Box Testing

Grey box testing is the combination n of black box and white box testing. Intention of this testing is to find out defects related to bad design or bad implementation of the system.

Testing Used For Web Based Application

This is done for 3 tier applications (developed for Internet / intranet /Extranet).Here we will be having Browser, web server and DB server. The applications accessible in browser would be developed in HTML, DHTML, XML, JavaScript etc. (We can monitor through these applications)

5.7. Level Of Testing Used In Project

5.7.1. Unit Testing

Initialization testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable**.**

5.7.2.Inegration Testing

All module which make application are tested . Integration testing is to make sure that the interaction of two or more components produces results that satisfy functional requirement**.**

5.7.3.System Testing

To test the complete system in terms of functionality and non functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

5.7.4.Functional Testing

The outgoing links from all the pages from specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields. Wrong inputs to the fields in the forms.

5.7.5.Alpha Testing

Alpha testing is final testing before the software is released to the general public. This testing is conducted at the developer site and in a controlled environment by the end user of the software.

5.7.6.Beta Testing

The beta test is conducted at one or more customer sites by the end user of the software. The beta test is conducted at one or more customer sites by the end user of the software.

5.7.7.Unit Testing Cases

Initialization testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable**.**

5.7.8.System Testing

To test the complete system in terms of functionality and non functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

5.7.9.Functional Testing

It is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing).

5.7.10.Integration Testing

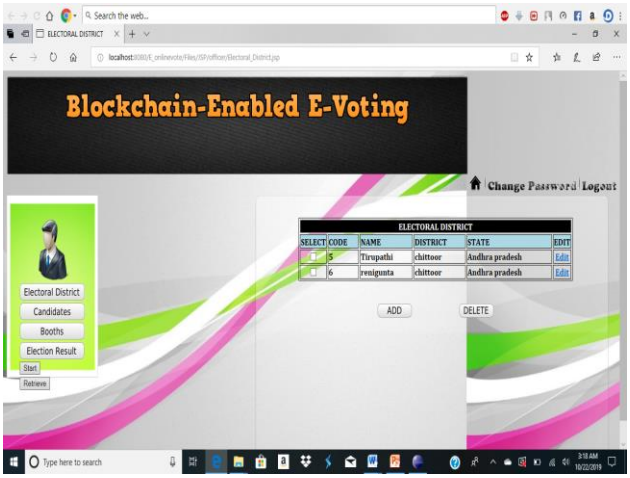
The outgoing links from all the pages from specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields. Wrong inputs to the fields in the forms.

Screen Shots

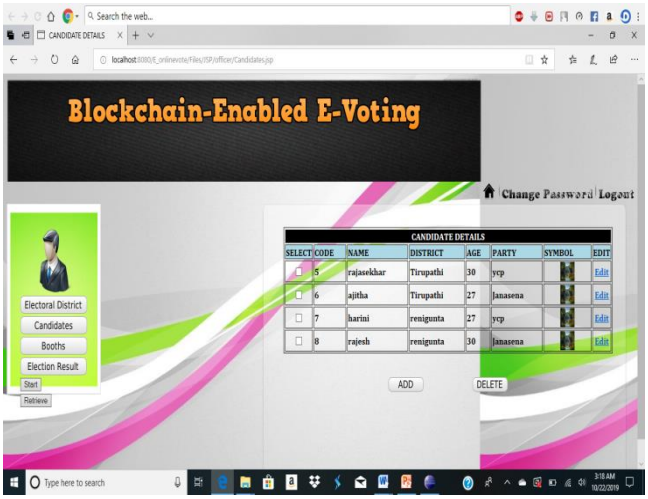
Login page

When we enter the address of local host, the home page will be displayed as shown below:

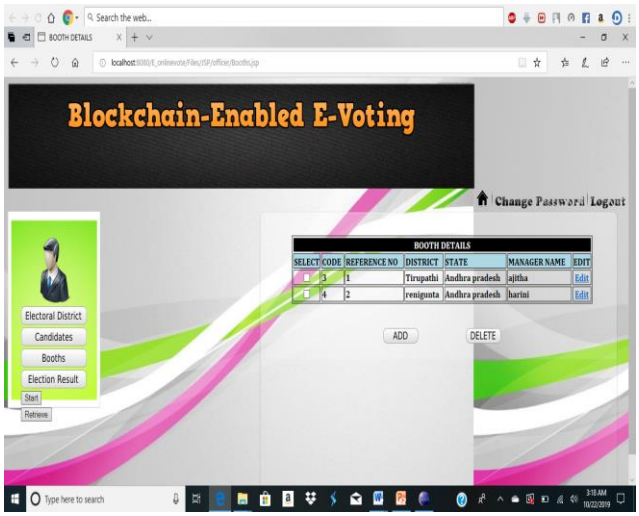
Electoral District



Candidate Details

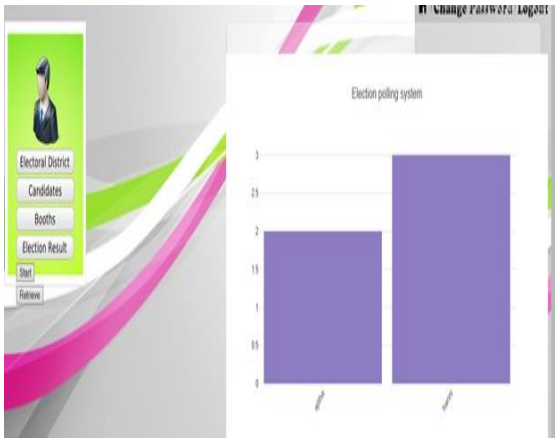


Booth Details



Election Result

Graph Result



Voting Process

VI. CONCLUSION

Block chain technology is currently in a nascent state. There haven't been enough distributed ledger-technology and Block chain-based applications to sufficiently evaluate whether this technology is superior to current voting systems. No full implementation of Block chain based E-Voting (BEV) for a national election has occurred. However, we argue that BEV has a future in elections and might transform voting. BEV can ensure security and transparency and reduce electoral violence. We using Fingerprint for fake voter's identification. This has faster authentication. It can also produce more mathematically accurate election results. Because BEV doesn't require management from a central authority, voting related costs will decrease. Finally, BEV should reduce the cost of paper based elections and increase voter participation. By this system achieving more votes and the votes are recorded accurately, permanently, securely and transparently.

REFERENCES

[1] J. Demuro, "Here Are the 10 SectorsThat Blockchain Will Disrupt Forever,"TechRadar Pro, 16 Jan. 2018.
[2] B. Dickson, "Blockchain Tech CouldFight Voter Fraud—and These CountriesAre Testing It," VentureBeat,22 Oct. 2016.
[3] J. Hall, "Can Blockchain TechnologySolve Voting Issues?," *Bitcoin Magazine*,7 Mar. 2018.
[4] A. Sandre, "Blockchain for Votingand Elections," Hackernoon, 14 Jan.2018.

[5] G. Prico, "Sierra Leone PilotsBlockchain-Based Voting for Political Elections," 22 Mar. 2018.