

**SCHOOL OF BUSINESS AND ECONOMICS**

**DEPARTMENT OF BUSINESS TECHNOLOGY**

**COURSES:** SYSTEMS ENGINEERING, DATABASE AND JAVA PROGRAMMING

**Project Name:**

**ONLINE VOTING SYSTEM**

**Student name: HAGENAYEZU Moise**

**Reg No: 222006319**

**Submitted to: Dr. BUGINGO Emmanuel**

**Date: 09/ 03/ 2024**

TABLE O

Contents

1. Introduction………………………………………………………………………………5

[2. Statement of the problem 6](#_Toc138833734) [Description of the existing system 6](#_Toc138833735)

[2.2 Problems of the current system 6](#_Toc138833736)

[2.3 How the proposed system will work 7](#_Toc138833737)

[2.4 Describe other alternatives 7](#_Toc138833738)

[3. Software requirement 7](#_Toc138833739)

[3.1 User Requirements 8](#_Toc138833740)

[3.1.1 Functional requirement 8](#_Toc138833741)

[3.1.2 Non-function requirement](#_Toc138833742)

[Portability requirement 8](#_Toc138833743)

[Reliability requirement 8](#_Toc138833744)

[Usability requirement 8](#_Toc138833745)

[Space requirement 8](#_Toc138833746)

[Organization requirement 8](#_Toc138833747)

[Implementation requirement 8](#_Toc138833748)

[External environment requirement 8](#_Toc138833749)

[Privacy requirement 8](#_Toc138833750)

[Safety requirement 8](#_Toc138833751)

[3.2 System requirements 8](#_Toc138833752)

[3.2.1 Minimum End-user Hardware Requirements 8](#_Toc138833753)

[3.2.2 Minimum End-user Software Requirements 8](#_Toc138833754)

[3.3 Software specifications 8](#_Toc138833755)

[**4.** DATA FLOW DIAGRAM (LEVEL 0, LEVEL 1**)** 9](#_Toc138833756)

[4.1 DFD LEVEL 0 9](#_Toc138833757)

[4.2 DFD LEVEL 1 10](#_Toc138833758)

[5. Feasibility study 10](#_Toc138833759)

[5.1 Technical Feasibility 10](#_Toc138833760)

[5.2 Financial Feasibility 10](#_Toc138833761)

[5.3 Market Feasibility 11](#_Toc138833762)

[5.4 Economic Feasibility 11](#_Toc138833763)

[5.5 Legal and Regulatory Feasibility 11](#_Toc138833764)

[5.6 Operational Feasibility 11](#_Toc138833765)

[6. Description of the selected Process Model 11](#_Toc138833766)

[6.1 Cause of your selection 11](#_Toc138833767)

[7. Conclusion 11](#_Toc138833768)

[8. References 11](#_Toc138833769)

CHAPTER2: DATABASE DESIGN ... ……………………………………….15

2.1 INTRODUCTION ………………………………………………………………………15 2.2 SECTION1: ... ………………………………………………………………………….16

# 2.2.1 ENTITIES ... ………………………………………………………………………………………….16

*2.2.2 LOGICAL DATABASE MODEL …………………………………………………………………21*

1. *2.2ENTITY RELETIONSHIP DIAGRAM ……………………………………………………………22* 
   1. SECTION2:SQL ….……………………………………………………………………24 2.4 SECTION3 ………………………………………………………………………………26

*2.4.1 VIEWS, TRIGGERS AND PROCEDURES ………………………………………………….26*

* 1. CONCLUSION ……………………………………………………………………...........39

CHAPTER3: JAVA PROGRAMMING ....................................................................................40

## 3.1 INTRODUCTION ………………………………………………………………………...40 3.2 TOOLS USED TO DEVELOP THIS SYSTEM IN JAVA PROGRAMMING …………40 3.3 FORMS DESCRIPTION …………………………………………………………………41

## 3.4 SPECIAL BUTTONS ...……………………………………………………………………52 3.5 CONCLUSION …………………………………………………………………………....52

**TOPIC:**



# Introduction

The winners of an election are usually satisfied with the outcomes, but it is often more challenging to the side of the losers and their supporters with the followers that they lost the election competition. To that end, it is not sufficient that election process and results are accurate. The public must also know the results are accurate, which can only be achieved if conduct of the election is sufficiently transparent that candidates, the press, and the general public can satisfy themselves that no errors or cheating have occurred.

# Statement of the problem

The problems that are related to this system are so many depending on how the situation we have met in previous years in several countries around the word. This is because there were some case of cheating to between those who are competing to win the selection. This Electronic Voting System has been developed to help eliminate any chance of tampering and improve the reliability and dependability of a voting system.

## 2.1 Description of the existing system

Voting is a method for a group of individuals to make a collective decision or voice their cumulative opinion to arrive at a consensus. The results of a vote can have far reaching consequences due to which it is imperative to maintain integrity by ensuring that there is no scope for fraud or cheating to occur while the votes are being cast. It is highly crucial that voting is carried out in a fair and just manner. The existing problems which are commonly identified are:

a. insufficient equipment that are used during the election process.

b. using unqualified workers during the election process

c.it took too much time for completing the action due to high number of people who are on the lines waiting to enter in the election room

## 2.2 Problems of the current system

The current system of voting, depending on the specific region or country, may have various problems and challenges. Here are some common issues associated with traditional voting systems:

Limited Voting Options: Traditional voting systems typically offer limited options for voters, such as voting on a single day or during specific hours.

Potential for Errors and Fraud: Traditional voting systems are susceptible to human errors, such as manual vote counting mistakes or illegible ballots. Moreover, they may also be vulnerable to fraud, including vote tampering, impersonation, or ballot stuffing, which can undermine the integrity of the election process

Long Queues and Wait Times: During elections, long queues and wait times at polling stations are common, leading to inconvenience for voters. This can discourage some individuals from participating in the voting process, potentially impacting the overall voter turnout.

## 2.3 How the proposed system will work

# In this system, the admin is the sole authority to manage elections, candidates and voters. Admin can also view the votes. Admin can also check if any vote is tampered, thus checking and verifying the block.

# Voter can view Elections and cast their vote, also can view the winner but cannot see the winning ratio or votes etc. The system uses Blockchain technology to create a block of every vote thus protecting its identity.

## Describe other alternatives

There are several alternative approaches to developing an online voting system. Here are a few commonly considered options:

Hybrid Voting Systems: Hybrid systems combine both electronic and traditional paper-based voting methods. They provide the flexibility for voters to choose between casting their vote electronically or using traditional methods, such as paper ballots. This approach addresses concerns about cybersecurity risks while still incorporating elements of convenience and efficiency offered by electronic voting.

Blockchain-Based Voting Systems: Blockchain technology can be utilized to create a secure and transparent online voting system. Blockchain provides a decentralized and immutable ledger that can enhance security, data integrity, and transparency in the voting process. Each vote is recorded as a transaction on the blockchain, making it tamper-proof and auditable.

Mobile Voting Apps: Mobile voting apps leverage smartphone technology to enable voters to cast their ballots electronically. These apps provide a user-friendly interface, identity verification mechanisms, and secure data transmission to ensure the integrity of the voting process. However, careful attention must be given to security measures and privacy concerns.

Remote Internet Voting: Remote internet voting allows voters to cast their ballots from any location with an internet connection. This approach eliminates the need for physical polling stations, providing convenience and accessibility to voters. However, it poses significant challenges in terms of security, authentication, and ensuring the secrecy of the vote.

In-Person Electronic Voting: In-person electronic voting systems involve using electronic voting machines at designated polling stations. These machines provide a digital interface for voters to cast their ballots securely. They offer the advantages of fast and efficient vote counting, but proper security measures must be in place to prevent tampering or hacking attempts.

## Software requirement

This section describes how the project is explained and how the software requirements must be organized in a requirement documents. In a software system, a requirement is defined as a condition that must be processed by a system to satisfy a user’s need, standard, specification, or other formally imposed documents. Software requirements that can be needed in this system are:

1. Visual studio 2019
2. SQL Server Management Studio
3. Web browser

## 3.1 User Requirements

The user requirement is defined as the statements that defines the full information about how the system work in real natural human language which will be easily understood by the user, and it will be without any technical terms that can be complicated. The user requirement should describe the functional and non-functional requirements so that they are understandable by system user who do not have detailed technical knowledge.

### 3.1.1 Functional requirements

Functional requirements act as the statements of services that the system should provide and its functionalities. It may be high level statements of how the system should do. The functional requirements of this system project comprise of 2 major modules with their sub-modules as follows:

Admin:

* Login:
  + Admin can login using id and password.
* Manage Elections:
  + Add/update/delete/view Elections.
* Manage Candidates:
  + Add/update/delete/view Candidates.
* Manage Voters:
  + Add/update/delete/view Voters.
* View Votes:
  + list of elections.
  + List of voters
  + Winner
  + Verify data with block created to know if its tampered

Voter:

* Login:
  + User can login in his personal account using id and password.
* Profile:
  + User can view and update his profile.
* Change Password:
  + can change the password within app
* Elections:
  + Pending / Participated
  + View elections
  + View Candidates
  + Cast Vote if not
  + Winner will be shown if election is complete

### 3.1.2 Non-function requirement

This requirement can also be called quality requirement. It describes the system properties which is not concerned with specific services that be delivered by the system. It is almost seen that the non-functional requirements are more critical than functional ones but my system will fulfill the end-users desire as it will be developed and been tested in all its functions before being implemented and been published in the society. The attributes that are related to the non- functional requirements are such as reliability, efficiency, usability, maintainability and portability of the system. Some of them are listed below:

* + - 1. The online voting system should accommodate may users during the election process of any organization/ committee at any time of the event.
      2. The system should display the confirmation message to the user within the short time (within few seconds) after his/her submission of the required information to the system.
      3. This system should be available to all organizations both private and public organizations and works twenty-four hours per day in seven days.

### Portability requirement

Portability requirement determines how our system or its elements can be launched within one environment or another. Online voting system should include software, hardware and other usage platform specifications.

### Reliability requirement

The reliability requirements for an online voting system project may vary depending on the specific context and the needs of the stakeholders involved. However, here are some common reliability requirements that are typically considered in such projects:

Availability: The online voting system should be available to voters and election administrators during the designated voting period. It should have a high uptime and minimal downtime to ensure that voters can access the system and cast their votes without interruptions.

Fault tolerance: The system should be designed to handle failures and errors gracefully. It should be able to recover from hardware or software failures, network issues, or other unforeseen problems without losing any votes or compromising the integrity of the election.

Data integrity: The online voting system should ensure the integrity of the votes cast. This includes preventing unauthorized modifications to the votes and ensuring that each vote is accurately recorded and counted. Robust encryption and secure transmission protocols should be used to protect the integrity of the data.

Security: The system should have strong security measures in place to protect against unauthorized access, tampering, or manipulation of the voting process.

### Usability requirement

For the online voting system, its crucial to prioritize usability to ensure that voters can easily navigate and participate in process. Here are some usability requirements to consider on this system project:

1. It will be easy to learn and understand
2. It will be efficient to use and making the user enjoy its usage and advantages
3. Being easy to remember its operations
4. This system will meet with few errors at low percentage
5. The online voting system will be subjectively satisfying the user’s need
6. Redundancy and High Availability: Implement redundancy and high availability measures to ensure continuous operation of the online voting system. This may involve having backup servers, failover systems, and load balancing to handle high traffic and minimize downtime.
7. User Training and Awareness: Provide comprehensive training and awareness programs for users, including election officials, system administrators, and voters. Educate them about safe computing practices, the risks associated with online voting, and how to identify and report any suspicious activities or potential security breaches.

### Space requirement

The space requirements for an online voting system can vary depending on various factors, such as the number of users, the complexity of the system, and the technology infrastructure. Here are some key considerations regarding space requirements for our system project:

Server Space: The online voting system will be fulfilled server space to store and manage the necessary data, including user information, voting records, and system configurations. The amount of server space needed will depend on the anticipated number of voters and the size of the data being stored.

Database Storage: The voting system will have a database to store and retrieve user profiles, authentication data, and voting records. The size of the database will depend on the number of registered voters and the amount of data associated with each user, such as personal information and voting history.

Bandwidth: Sufficient bandwidth is crucial to handle the incoming and outgoing data traffic generated by the online voting system. The required bandwidth will depend on the expected number of concurrent users, the size of the data being transmitted (e.g., ballot information, voting selections), and the frequency of user interactions.

Backup and Redundancy: It's important to allocate space for regular backups and redundant systems to ensure data integrity and system availability. This includes storing backup copies of the database, server configurations, and other critical system components.

Physical Infrastructure: In addition to digital space requirements, consider the physical infrastructure needed to support the online voting system. This may include server racks, cooling systems, power supply units, networking equipment, and physical security measures.

### Organization requirement

The organizational requirements are often requirements for the users that originate from the requirements for the interactive system or that lead to requirements for the interactive system. Online voting system should require for networking, collaborating, and sharing such as the capacity of mutual trust, the capacity of enabling and the capacity of enacting. Here are some organizational requirements to consider for an online voting system:

Clear Roles and Responsibilities: Define clear roles and responsibilities for each team member involved in the online voting system project. This helps to ensure accountability and efficient collaboration among team members.

Legal Compliance: Ensure compliance with relevant laws and regulations governing the electoral process and data protection. Understand the legal requirements for online voting systems, such as voter eligibility, data privacy, security measures, and auditing.

Stakeholder Engagement: Identify key stakeholders, such as election officials, government authorities, and relevant organizations, and establish clear communication channels to engage with them throughout the project. Seek their input, address concerns, and keep them informed about system developments and updates.

Training and Education: Provide comprehensive training and educational materials to relevant personnel involved in managing and operating the online voting system. This includes training election officials, system administrators, support staff, and volunteers on system usage, security protocols, troubleshooting, and voter assistance.

Communication Plan: Develop a comprehensive communication plan to inform voters about the online voting system, including its availability, registration process, voting instructions, and support channels. Utilize various communication channels, such as websites, social media, press releases, and direct outreach, to reach out to voters.

### Implementation requirement

Implementing an online voting system requires careful planning and execution to ensure a successful deployment. Here are some implementation requirements to consider for an online voting system project:

System Architecture: Define the overall system architecture of the online voting system, including the front-end user interface, back-end infrastructure, database structure, and integration with any existing systems or databases. Determine the technology stack and frameworks that will be used for development.

Security Measures: Implement robust security measures to protect the integrity and confidentiality of the online voting system. This includes secure user authentication mechanisms, encryption of sensitive data, protection against cyber threats (e.g., DDoS attacks, hacking attempts), and adherence to established security standards.

Voter Registration: Develop a user-friendly voter registration process that allows eligible voters to create accounts and authenticate their identities securely. Implement appropriate validation checks and verification methods to ensure the accuracy and integrity of voter information.

Ballot Creation and Management: Design a system that enables the creation, management, and distribution of ballots for different elections or voting events. Consider features such as multiple ballot formats (e.g., single-choice, ranked-choice), candidate or issue information, and ballot customization options.

Voter Authentication and Authorization: Implement robust mechanisms for voter authentication and authorization to ensure that only eligible voters can access and participate in the online voting system. Use secure login credentials, two-factor authentication, and voter verification methods as required by the jurisdiction.

Results Calculation and Reporting: Design algorithms and processes to accurately calculate and tally the voting results. Develop reporting features that provide transparent and accessible election results to stakeholders, such as election officials, candidates, and the public.

User Support and Training: Provide comprehensive user support materials, including user guides, FAQs, and tutorials, to assist voters in using the online voting system. Conduct training sessions for election officials, system administrators, and support staff to ensure they are equipped to handle user inquiries and technical issues.

### External environment requirement

When implementing an online voting system, it's important to consider the external environment in which the system will operate. Here are some external environment requirements to consider for an online voting system:

Legal and Regulatory Compliance: Ensure that the online voting system complies with all relevant laws, regulations, and guidelines governing elections, data protection, privacy, and cybersecurity. Understand the legal requirements and obtain necessary approvals or certifications to operate the system within the jurisdiction.

Public Trust and Confidence: Foster public trust and confidence in the online voting system by being transparent about the system's security measures, data protection practices, and auditability. Communicate the benefits, safeguards, and integrity of the system to stakeholders, including voters, political parties, and civil society organizations.

Collaboration with Election Authorities: Collaborate closely with election authorities or relevant government bodies responsible for managing elections. Seek their expertise, guidance, and cooperation to ensure that the online voting system aligns with their requirements and integrates seamlessly into the overall electoral process.

Infrastructure Readiness: Assess the existing infrastructure, such as internet connectivity, network reliability, and power supply, to determine if it can support the online voting system. Identify any necessary upgrades or improvements needed to ensure the system's accessibility and availability.

Stakeholder Engagement: Engage with stakeholders, including political parties, advocacy groups, and community organizations, to gather feedback, address concerns, and incorporate diverse perspectives into the design and implementation of the online voting system. Involve stakeholders in the decision-making processes to enhance system acceptance and inclusivity.

International Standards and Best Practices: Stay informed about international standards, best practices, and case studies related to online voting systems. Learn from the experiences of other jurisdictions and adopt proven practices to enhance the security, usability, and integrity of the system.

Continuous Monitoring and Evaluation: Implement mechanisms for continuous monitoring and evaluation of the online voting system's performance. Regularly assess the system's effectiveness, security, and user satisfaction. Conduct audits or independent reviews to validate the integrity of the system and identify areas for improvement.

Crisis and Contingency Planning: Develop crisis and contingency plans to address potential disruptions, such as cyber threats, natural disasters, or system failures. Establish backup systems, data recovery processes, and communication protocols to ensure the continuity of the online voting system in unforeseen circumstances.

### Privacy requirement

### Ensuring the privacy of voters is of utmost importance in an online voting system. Here are some key privacy requirements to consider for an online voting system project:

### Data Protection Compliance: Ensure compliance with applicable data protection laws, regulations, and guidelines. Understand the legal requirements related to the collection, storage, processing, and sharing of voter data, and implement necessary safeguards to protect personal information.

### Secure Data Transmission: Implement strong encryption protocols (e.g., SSL/TLS) to ensure the secure transmission of voter data between users' devices and the online voting system servers. This protects the privacy and integrity of the data while it is in transit.

### Anonymity: Design the online voting system in a way that preserves voter anonymity. Implement mechanisms to separate voter identities from their votes, ensuring that votes cannot be traced back to individual voters. Consider methods such as encryption, blind signatures, or mix networks to protect anonymity.

### Confidentiality: Implement measures to ensure the confidentiality of voter data stored within the online voting system. Use secure storage systems, access controls, and encryption to protect against unauthorized access, disclosure, or manipulation of voter information.

### Minimized Data Collection: Limit the collection of personal data to only what is necessary for the functioning of the online voting system. Minimize the use of personally identifiable information and avoid collecting unnecessary or sensitive data that could compromise privacy.

### Informed Consent: Obtain explicit and informed consent from voters regarding the collection, storage, and use of their personal data for the purposes of the online voting system. Clearly communicate the data handling practices and provide options for voters to control and revoke their consent.

### Data Retention and Deletion: Establish policies and procedures for data retention and deletion. Define specific timeframes for retaining voter data, and ensure that personal data is securely deleted or anonymized once it is no longer necessary for the operation of the online voting system.

### Third-Party Processing: If any third-party services or vendors are involved in the online voting system, ensure that they adhere to robust privacy and security standards. Conduct due diligence to ensure that any data processing or storage performed by third parties aligns with privacy requirements and does not compromise voter privacy.

### Transparency and Auditability: Design the system to be transparent and auditable, allowing for independent verification of the privacy measures in place. Implement logging and auditing mechanisms to track system activities, monitor data access, and detect any unauthorized or suspicious behavior.

### User Control and Access Rights: Provide voters with control over their personal data. Offer options for voters to access, review, correct, and delete their data as required by privacy regulations. Implement appropriate authentication and authorization mechanisms to ensure that only authorized individuals can access or modify voter data.

### Privacy Policies and Notices: Develop and publish clear and comprehensive privacy policies and notices that explain how voter data is collected, used, stored, and protected within the online voting system. Make these policies easily accessible to voters to promote transparency and build trust.

### Privacy Impact Assessment: Conduct a privacy impact assessment to identify and mitigate potential privacy risks associated with the online voting system. Assess the impact of the system on voter privacy and implement measures to address any identified risks or vulnerabilities.

It's important to consult with legal experts, privacy professionals, and data protection authorities to ensure that the online voting system meets the privacy requirements specific to the jurisdiction in which it will be deployed. Privacy should be a fundamental consideration throughout the design, development, and operation of the system.Bottom of Form

### Safety requirement

Safety is a critical aspect of an online voting system to ensure the integrity of the voting process and protect against potential threats. Here are some key safety requirements to consider for an online voting system project:

System Security: Implement robust security measures to protect the online voting system from unauthorized access, tampering, or manipulation. This includes secure authentication mechanisms, encryption of data in transit and at rest, firewalls, intrusion detection systems, and regular security updates.

Resilience to Cyber Attacks: Design the system to withstand and recover from cyber-attacks, such as distributed denial-of-service (DDoS) attacks or attempts to compromise the integrity of the system. Implement measures like traffic monitoring, redundancy, disaster recovery plans, and incident response procedures.

Secure Infrastructure: Ensure that the infrastructure supporting the online voting system, including servers, networks, and databases, is physically and digitally secure. Implement appropriate access controls, monitoring systems, and physical security measures to prevent unauthorized access or tampering.

Data Integrity: Implement mechanisms to ensure the integrity of voting data throughout the entire process, including data transmission, storage, and processing. Use digital signatures, hashing algorithms, and checksums to verify the authenticity and integrity of data at various stages.

Auditing and Logging: Implement robust logging and auditing mechanisms to track system activities, detect any unauthorized access or suspicious behavior, and maintain an audit trail of system activities, including user actions and system changes.

Disaster Recovery and Business Continuity: Develop a comprehensive disaster recovery plan and business continuity strategy to address potential disruptions, such as system failures, natural disasters, or cyber attacks. Regularly backup data and test the recovery procedures to ensure the system can be restored in a timely manner.

Redundancy and High Availability: Implement redundancy and high availability measures to ensure continuous operation of the online voting system. This may involve having backup servers, failover systems, and load balancing to handle high traffic and minimize downtime.

User Training and Awareness: Provide comprehensive training and awareness programs for users, including election officials, system administrators, and voters. Educate them about safe computing practices, the risks associated with online voting, and how to identify and report any suspicious activities or potential security breaches.

System Testing and Penetration Testing: Conduct rigorous testing, including penetration testing, to identify and address vulnerabilities in the online voting system. Regularly perform security assessments and vulnerability scans to ensure the system's resilience against potential threats.

Collaboration with Security Experts: Engage with security experts, independent auditors, or third-party organizations to assess the security of the online voting system. Seek external input and audits to validate the effectiveness of security measures and identify potential weaknesses.

Incident Response and Communication: Establish an incident response plan to address security incidents, breaches, or disruptions promptly. Define clear procedures for reporting incidents, communicating with stakeholders, mitigating risks, and conducting post-incident analysis.

## 3.2 System requirements

The system requirements for an online voting system project can vary depending on specific needs and circumstances. However, here are some common system requirements that are typically considered in such projects: Web-Based Interface: The system should have a user-friendly web-based interface that allows voters to access the system easily and cast their votes securely. The interface should be intuitive, responsive, and accessible across different devices and browsers.

Authentication and Authorization: The system should support robust authentication mechanisms to verify the identity of voters and ensure that only eligible individuals can participate in the voting process. This may include username/password authentication, two-factor authentication, or biometric authentication, depending on the requirements and security considerations.

Security Measures: The online voting system should employ various security measures to protect the integrity and confidentiality of the voting process. This includes encryption of data transmission, secure storage of sensitive information, protection against unauthorized access or tampering, and regular security updates and patches to address vulnerabilities.

Vote Casting and Verification: The system should provide a secure and reliable method for voters to cast their votes electronically. It should also enable voters to verify that their votes have been accurately recorded and counted without compromising the anonymity of the voting process.

### 3.2.1 Minimum End-user Hardware Requirements

The minimum end-user hardware requirements for an online voting system project can vary based on the specific design and implementation of the system. However, here are some general guidelines for minimum hardware requirements:

**Desktop/Laptop Computers:**

Processor: Dual-core or higher

RAM: 2 GB or higher

Hard Disk Space: 10 GB or higher

Operating System: Windows, macOS, or Linux with a supported browser

Internet Connection: Broadband connection for a smooth online experience

**Mobile Devices**:

Smartphone or tablet with a supported operating system (e.g., iOS, Android)

Sufficient storage space for the voting app installation

Stable internet connection (Wi-Fi or mobile data)

It's important to consider that these are just minimum requirements and may vary depending on the complexity of the online voting system, the user interface, and the level of interactivity involved. It's also recommended to provide clear guidelines to users regarding supported browsers, minimum screen resolutions, and any additional software dependencies required to access and interact with the online voting system.

Additionally, considering accessibility requirements, it's essential to ensure that the online voting system is compatible with assistive technologies, such as screen readers, for users with disabilities. This may involve adherence to accessibility standards and guidelines, including WCAG (Web Content Accessibility Guidelines).

When developing an online voting system, it's advisable to conduct user testing on a variety of hardware configurations to ensure a smooth user experience across different devices and platforms.

### 3.2.2 Minimum End-user Software Requirements

The minimum end-user software requirements for an online voting system project can vary based on the specific technology stack and design choices made for the system. Here are some general considerations for the minimum software requirements:

Web Browsers:

Commonly used web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, or Opera.

The system should be compatible with the latest stable versions of these browsers, as well as some previous versions to accommodate a wider range of users.

Operating Systems:

Windows: The system should be compatible with recent versions of Windows, such as Windows 10, Windows 8, or Windows 7.

macOS: The system should be compatible with recent versions of macOS, such as macOS Big Sur, macOS Catalina, or macOS Mojave.

Linux: The system should be compatible with popular distributions like Ubuntu, Fedora, or CentOS, with corresponding browser support.

## Software specifications

The software specifications for an online voting system project can vary depending on the specific requirements and design choices made for the system. However, here are some common software specifications to consider:

Programming Language and Framework:

The choice of programming language and framework depends on the development team's expertise and the project's specific needs. Common languages and frameworks used for web development include Python (Django, Flask), Java (Spring), Ruby (Ruby on Rails), or JavaScript (Node.js, React, Angular).

Web Server:

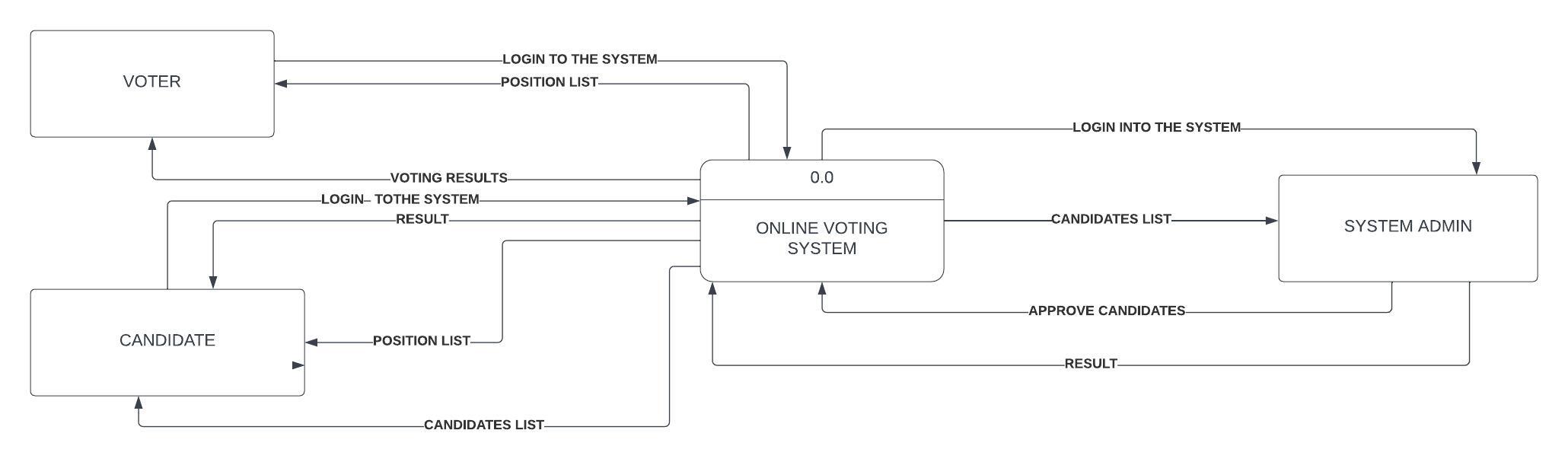
The online voting system will require a web server to host the application and handle incoming requests from users. Common web servers include Apache HTTP Server, Nginx, or Microsoft IIS.

Database Management System:

A database management system (DBMS) is required to store and manage the data related to the voting system, such as user information, candidate details, and voting records. Popular options include MySQL, PostgreSQL, Oracle Database, or MongoDB (for NoSQL databases).

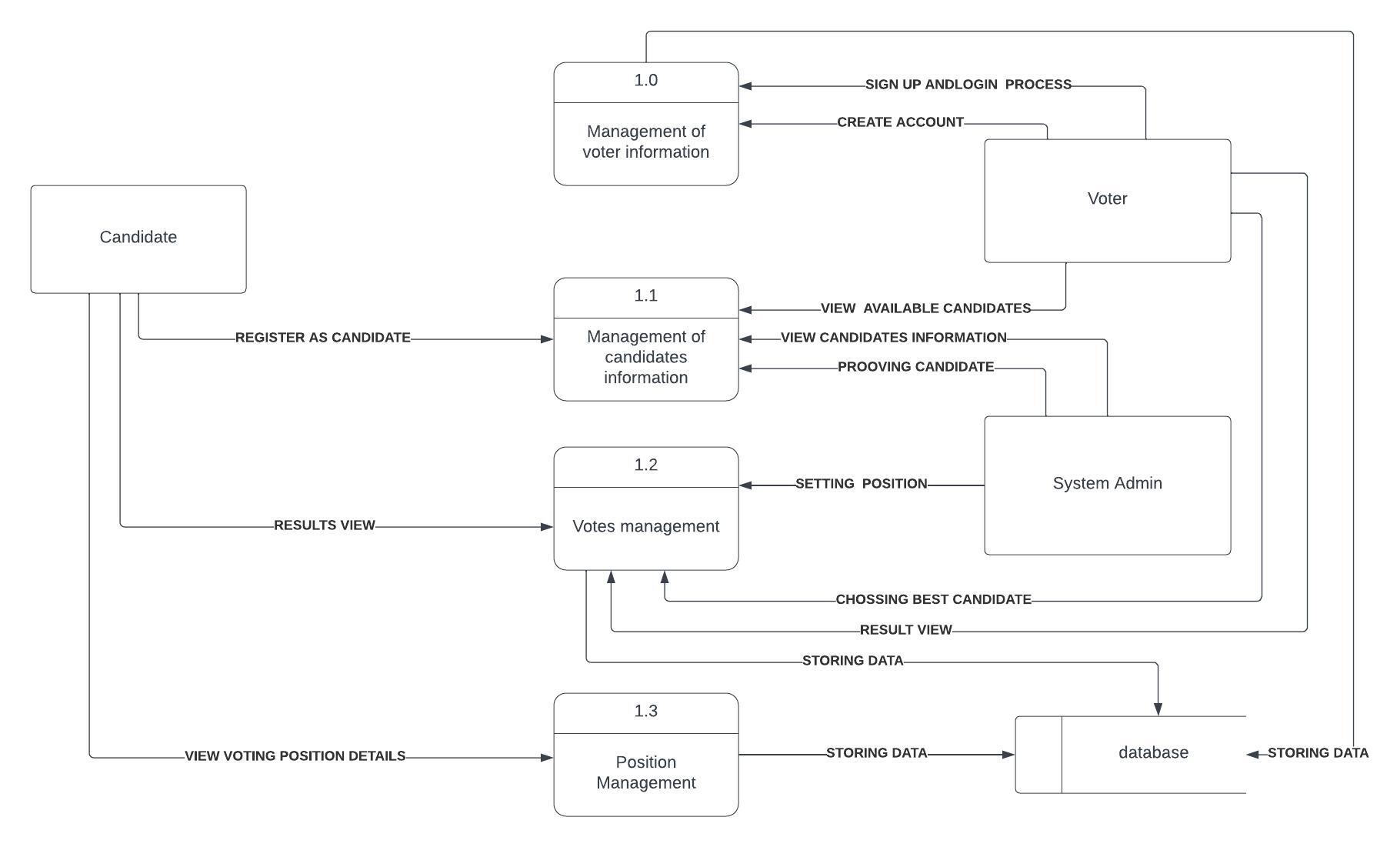
# DATA FLOW DIAGRAM (LEVEL 0, LEVEL 1**)**

4.1 DFD LEVEL 0



This figure defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it, this diagram is a high level view of a system.

## 4.2 DFD LEVEL 1



This diagram shows the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system.

# 5. Feasibility study

5.1 Technical Feasibility

The technical feasibility of an online voting system project depends on several factors. Here are some key considerations:

The technical feasibility of an online voting system project depends on several factors. Here are some key considerations:

Infrastructure: An online voting system requires a robust and scalable infrastructure, including servers, databases, networking, and security measures. The system should be able to handle a large volume of concurrent users and maintain uptime and performance during peak periods, such as during elections.

Security: Ensuring the security of the online voting system is crucial to prevent tampering, fraud, and unauthorized access. The system should employ strong encryption techniques to protect voter information and use secure protocols for data transmission. Additionally, measures like multi-factor authentication, firewalls, intrusion detection systems, and regular security audits should be implemented.

Authentication and Verification: An online voting system must have a reliable method to authenticate and verify the identity of voters. This may involve integrating with existing identification systems, such as government-issued ID cards or biometric systems, to ensure that each voter is unique and eligible to participate.

Privacy: Voter privacy is of utmost importance in any voting system. The online voting system should ensure that individual votes remain anonymous and cannot be linked to specific voters. Strong data protection measures, including anonymization techniques and strict access controls, should be implemented.

Accessibility: The online voting system should be accessible to a wide range of users, including those with disabilities or limited technological proficiency. It should comply with accessibility standards and provide features like screen readers, keyboard navigation, and language options to cater to diverse user needs.

Testing and Validation: Rigorous testing and validation processes are essential to identify and address any vulnerabilities or issues in the system. Thorough testing should be conducted to simulate real-world scenarios, including stress testing to evaluate system performance under high loads.

Legal and Regulatory Compliance: Online voting systems must adhere to legal and regulatory requirements specific to the jurisdiction in which they are implemented. Compliance with data protection laws, election regulations, and auditing requirements is necessary.

Contingency Planning: The online voting system should have contingency plans in place to address potential technical failures, power outages, or cyberattacks. Regular backups, redundancy measures, and disaster recovery plans should be implemented to ensure the system's resilience.

It is important to note that implementing an online voting system is a complex undertaking, and various challenges exist, including technological, security, and societal factors. Thorough planning, expertise in system design, and ongoing monitoring and improvement are crucial for the successful implementation of such a system.

## 5.2 Financial Feasibility

The financial feasibilities for an online voting system project involve evaluating the financial viability and sustainability of the project. Here are some key considerations:

Cost Estimation: Conduct a comprehensive cost estimation for the online voting system project. Identify and quantify the various costs involved, including system development, infrastructure setup, security measures, staff training, maintenance, and ongoing operational expenses. Consider both one-time and recurring costs.

Budgeting and Funding Sources: Develop a detailed budget for the project, allocating funds to different components and phases. Identify potential funding sources, such as government grants, public-private partnerships, or crowdfunding initiatives. Evaluate the availability and feasibility of accessing these funding sources to support the financial requirements of the project.

Cost-Benefit Analysis: Perform a cost-benefit analysis to assess the financial impact and potential benefits of implementing the online voting system. Consider both tangible and intangible benefits, such as cost savings, increased efficiency, improved accuracy, enhanced democratic participation, and reduced logistical expenses. Compare the projected benefits against the estimated costs to determine the financial feasibility of the project.

Return on Investment (ROI): Evaluate the potential return on investment for the online voting system project. Calculate the expected financial returns over a defined period and compare them with the initial and ongoing costs. Consider factors such as cost savings, increased revenue generation potential, improved resource allocation, and reduced risks. A positive ROI indicates the financial viability of the project.

Revenue Generation Opportunities: Explore revenue generation opportunities associated with the online voting system. This may include providing ancillary services related to the system, offering consulting or customization services, or licensing the technology to other electoral bodies or governments. Identify potential revenue streams and assess their financial viability.

Cost Reduction Potential: Assess the potential for cost reduction compared to traditional voting methods. Identify areas where the online voting system can lead to savings, such as reduced printing and distribution costs, decreased staffing requirements at polling stations, and shorter election result processing times. Quantify the potential cost reductions to determine the financial benefits.

Long-Term Sustainability: Evaluate the long-term financial sustainability of the online voting system. Consider ongoing maintenance, system upgrades, and adaptation to evolving technologies and security standards. Ensure that the project's financial plan accounts for these long-term requirements to avoid financial challenges in the future.

Financial Risk Management: Identify potential financial risks associated with the project and develop risk mitigation strategies. Consider risks such as budget overruns, changes in funding availability, unexpected expenses, or revenue shortfalls. Implement financial risk management measures to minimize the impact of these risks on the project's financial feasibility.

Cost of Compliance: Consider the financial implications of complying with legal and regulatory requirements. This includes costs associated with data protection, privacy measures, cybersecurity, and adherence to electoral laws. Ensure that the project's budget accounts for these compliance costs to avoid financial setbacks.

Cost Recovery and Payback Period: Evaluate the potential for cost recovery and calculate the payback period for the project. Determine how long it will take to recoup the initial investment and start generating positive financial returns. Consider factors such as revenue generation, cost savings, and other financial benefits to assess the project's payback period.

By conducting a thorough financial feasibility analysis and considering these factors, you can assess the financial viability and sustainability of an online voting system project.

## 5.3 Market Feasibility

The market feasibility of an online voting system project depends on various factors and can vary based on the specific context and location. Here are some key considerations to assess the market feasibility: Technological Infrastructure: Online voting systems require a robust technological infrastructure, including secure servers, encryption protocols, and reliable internet connectivity. Assessing the availability and quality of such infrastructure in the target market is crucial.

Legal and Regulatory Environment: The legal and regulatory framework surrounding elections and voting systems play a significant role. Investigate the laws and regulations related to online voting in the target market, including data protection, privacy, and electoral laws. Understanding the legal requirements and potential barriers is essential.

Trust and Security: Building trust is vital for the success of an online voting system. Public perception regarding the security and integrity of the system is crucial. Conducting thorough security audits, implementing robust authentication measures, and using encryption techniques can help address concerns about system vulnerabilities.

User Adoption and Acceptance: Evaluate the willingness of voters and relevant stakeholders to adopt online voting. Conduct market research, surveys, and focus groups to understand public opinion, concerns, and acceptance levels. Public awareness campaigns and education initiatives may be necessary to increase acceptance.

Accessibility and Inclusivity: Consider the accessibility of the online voting system to all segments of the population, including those with disabilities, elderly individuals, and those with limited technology access. Ensuring that the system is user-friendly, supports multiple languages, and accommodates various accessibility requirements can enhance its market feasibility.

Cost and Return on Investment: Assess the financial implications for implementing an online voting system. Consider the initial setup costs, maintenance expenses, and ongoing operational costs. Additionally, evaluate the potential benefits, such as reduced logistics expenses, increased voter participation, and streamlined electoral processes.

Political and Stakeholder Support: Gauge the level of political will and stakeholder support for implementing online voting. Engage with government officials, electoral bodies, political parties, and advocacy groups to understand their perspectives and potential collaborations.

Pilot Projects and Case Studies: Examine successful online voting system implementations in similar markets or regions. Analyze their outcomes, challenges faced, and lessons learned. Case studies and pilot projects can provide valuable insights into the market feasibility of an online voting system.

It's essential to conduct a comprehensive feasibility study that considers these factors specific to your target market to assess the viability of an online voting system project.

## 5.4 Economic Feasibility

The economic feasibility of an online voting system project involves assessing the financial viability and potential economic benefits. Here are some key considerations for evaluating the economics feasibility:

Cost-Benefit Analysis: Conduct a thorough cost-benefit analysis to evaluate the financial implications of implementing an online voting system. Consider the costs associated with system development, infrastructure setup, security measures, staff training, maintenance, and ongoing operational expenses. Compare these costs against the potential benefits, such as reduced printing and logistical costs, increased voter turnout, and streamlined election processes.

Return on Investment (ROI): Assess the potential ROI of the online voting system project. Calculate the expected financial benefits over a defined period and compare them with the initial and ongoing costs. Consider both tangible and intangible benefits, such as cost savings, increased efficiency, improved accuracy, and enhanced democratic participation.

Scalability and Cost Efficiency: Evaluate the scalability of the online voting system. Assess whether the system can handle a growing number of voters and elections without significant increases in costs. A scalable system can be cost-efficient in the long run, as it reduces the need for extensive infrastructure upgrades or redevelopments.

Cost Reduction Potential: Identify areas where the online voting system can potentially reduce costs compared to traditional voting methods. This may include savings in printing and distribution of paper ballots, reduced staffing requirements at polling stations, and shorter election result processing times.

Ancillary Services and Revenue Generation: Explore opportunities for generating additional revenue or providing ancillary services related to the online voting system. For example, offering consulting, customization, or maintenance services to other electoral bodies or governments. However, it's important to consider any legal and ethical considerations surrounding the provision of such services.

Long-Term Sustainability: Assess the financial sustainability of the online voting system beyond the initial implementation. Consider factors such as system maintenance, upgrades, and adaptation to evolving technologies and security standards. Ensure that the project's long-term financial requirements are adequately accounted for.

Funding Sources: Identify potential funding sources for the online voting system project. These may include government funding, public-private partnerships, grants, or crowdfunding initiatives. Assess the availability and feasibility of accessing these funding sources to support the implementation and sustainability of the project.

Cost of Potential Risks: Evaluate the potential risks associated with the online voting system project and their potential costs. These risks may include security breaches, technical failures, legal challenges, or public skepticism. Implementing appropriate risk mitigation measures can help minimize the financial impact of such risks.

By conducting a comprehensive economic feasibility analysis, considering these factors, and incorporating accurate cost estimates and projections, you can assess the viability and financial sustainability of an online voting system project.

## 5.5 Legal and Regulatory Feasibility

The legal and regulatory feasibilities for an online voting system project involve assessing the legal framework and compliance requirements related to elections and voting systems. Here are some key considerations:

Electoral Laws: Review the existing electoral laws and regulations in the target jurisdiction. Identify specific provisions related to voting methods, eligibility, voter registration, ballot secrecy, and the use of electronic systems. Ensure that the online voting system project aligns with the legal requirements and is permissible within the current legal framework.

Data Protection and Privacy: Assess the data protection and privacy laws applicable to the online voting system project. Consider requirements regarding the collection, storage, processing, and sharing of voter information. Ensure compliance with applicable data protection regulations, such as obtaining informed consent, implementing appropriate security measures, and protecting voter anonymity.

Authentication and Identity Verification: Evaluate the legal and regulatory requirements for voter authentication and identity verification in the online voting system. Verify the legality of the chosen methods, such as biometric authentication, digital signatures, or unique identification codes. Ensure compliance with relevant laws, such as those related to digital signatures or electronic identification.

Security and Integrity: Investigate the legal and regulatory standards for ensuring the security and integrity of the online voting system. Assess the requirements for encryption, secure transmission of data, protection against hacking or tampering, and auditability of the system. Compliance with relevant cybersecurity and IT regulations is crucial to maintain the legality and integrity of the online voting process.

Accessibility and Inclusivity: Consider legal obligations regarding accessibility and inclusivity in the electoral process. Assess requirements for accommodating voters with disabilities, language preferences, and limited technology access. Ensure that the online voting system project adheres to applicable laws related to accessibility and provides necessary support to all segments of the population.

Auditability and Transparency: Examine the legal requirements for auditability and transparency in the online voting system. Assess whether the system provides mechanisms for verifying the accuracy and integrity of the voting process, including the ability to audit the results and address any discrepancies. Compliance with relevant laws related to transparency and accountability is essential.

Certification and Compliance: Determine the legal requirements for certification and compliance of the online voting system. Some jurisdictions may have specific certification processes or standards that voting systems must meet. Ensure that the project adheres to these requirements and obtains necessary certifications or approvals from relevant authorities.

Legal Challenges and Precedents: Investigate any legal challenges or precedents related to online voting systems in the target jurisdiction or in similar contexts. Analyze court cases, legal opinions, and decisions that may impact the legality and feasibility of the project. Learn from past experiences and ensure that the project addresses potential legal concerns.

Consulting legal experts with expertise in election law, data protection, cybersecurity, and relevant legislation is crucial to ensure compliance with legal and regulatory requirements. It is important to stay updated on any changes or developments in the legal framework to maintain the feasibility and legality of the online voting system project.

## 5.6 Operational Feasibility

The operational feasibilities for an online voting system project involve assessing the practicality and effectiveness of implementing and managing the system. Here are some key considerations:

System Reliability and Performance: Evaluate the reliability and performance of the online voting system. Consider factors such as system uptime, response times, scalability, and the ability to handle a large volume of simultaneous users. The system should be robust enough to handle peak loads during elections without significant disruptions or slowdowns.

User-Friendliness: Assess the user-friendliness of the online voting system for both voters and election administrators. The system should have an intuitive interface, clear instructions, and easy navigation to ensure a positive user experience. Conduct user testing and gather feedback to identify and address any usability issues.

Training and Support: Evaluate the training and support requirements for stakeholders involved in the online voting system. Election administrators, IT personnel, and voters may require training on system usage, security protocols, and troubleshooting procedures. Adequate support channels should be available to address user queries or technical issues promptly.

Integration with Existing Systems: Consider the integration of the online voting system with existing electoral processes and systems. Assess compatibility with voter registration databases, identity verification systems, result tabulation systems, and other relevant components. Seamless integration can help streamline processes and reduce operational complexities.

Security Measures: Assess the effectiveness of security measures implemented in the online voting system. Consider encryption protocols, authentication mechanisms, data storage and transmission security, and vulnerability management. Regular security audits and adherence to best practices can help mitigate risks and ensure operational integrity.

Election Day Operations: Evaluate the operational processes and procedures required for conducting elections using the online voting system. This includes activities such as voter registration, ballot preparation, voter authentication, monitoring, and handling of technical issues. Develop comprehensive operational plans and protocols to ensure smooth execution on election day.

Risk Management: Identify potential risks and develop risk management strategies for the online voting system project. This includes addressing technical failures, cybersecurity threats, data breaches, and any legal or reputational risks. Implement contingency plans, backup systems, and disaster recovery protocols to minimize disruptions and mitigate potential risks.

Ongoing Maintenance and Updates: Consider the long-term operational requirements of the online voting system, including system maintenance, software updates, and ongoing technical support. Establish a framework for regular maintenance, bug fixes, and feature enhancements to ensure system stability and performance over time.

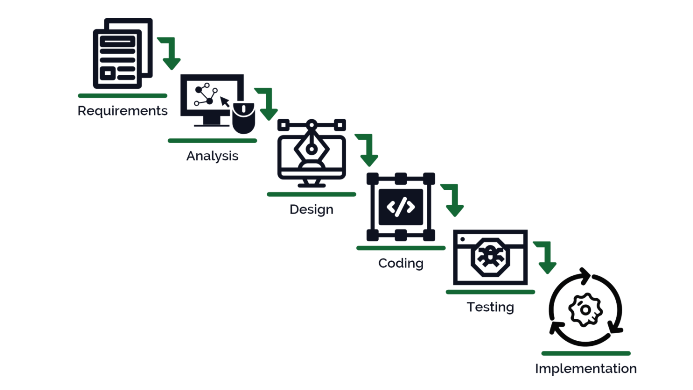
Pilots and Testing: Conduct pilot projects and extensive testing of the online voting system before full-scale implementation. This helps identify operational issues, validate system performance, and gather user feedback to fine-tune the system before actual elections.

Continuous Improvement: Establish mechanisms for continuous improvement and feedback gathering. Regularly assess the system's performance, collect user feedback, and make necessary adjustments based on lessons learned from each election cycle. This ensures that the online voting system evolves and adapts to changing needs and technological advancements.

By addressing these operational considerations, you can assess the feasibility and effectiveness of implementing and managing an online voting system project.

# Description of the selected Process Model

The selected process model for the development of an online voting system is the Waterfall model. The Waterfall model is a sequential approach to software development that proceeds linearly through defined phases. Here's a description of how the Waterfall model can be applied to the development of an online voting system:



## Cause of your selection

The selection of the waterfall process model for an online voting system project could be influenced by several factors. Here are some possible reasons why the waterfall model might be chosen:

Clarity of Requirements: If the requirements for the online voting system are well-defined and stable from the beginning, the waterfall model can be suitable. This model follows a sequential approach, where each phase is completed before moving on to the next. With clear and fixed requirements, it becomes easier to plan and execute each phase accordingly.

Linear and Predictable Process: The waterfall model offers a linear and predictable process flow, which can be advantageous for projects with a well-understood scope. In the case of an online voting system, where the stages like requirement analysis, design, development, testing, and deployment need to be executed in a predefined order, the waterfall model can provide a structured approach.

Regulatory or Legal Compliance: Certain projects, including online voting systems, often need to comply with specific regulatory or legal requirements. The waterfall model's documentation-heavy nature can aid in meeting these compliance standards. By documenting each phase thoroughly, including requirements, design, and testing, it becomes easier to demonstrate compliance during audits or reviews.

Limited Iterations and Changes: The waterfall model is known for its sequential nature, where each phase is completed before moving on to the next. It assumes minimal changes in requirements during the development process. If the online voting system project has stable requirements and limited scope for iterations, the waterfall model can be suitable.

Well-Defined Project Timeline and Budget: The waterfall model allows for better estimation of project timelines and budgets. With its linear approach, the phases and their durations can be planned in advance, which can aid in resource allocation and budgeting for the online voting system project.

It's important to note that the waterfall model may not be suitable if the project has evolving requirements, a need for frequent feedback and iterations, or a high level of uncertainty. In such cases, alternative models like Agile or iterative approaches might be more appropriate. The choice of the process model should always align with the specific characteristics and requirements of the project.

# Conclusion

# References

www.incose.org>system-engineering



**DATA BASE OF THE SYSTEM**

**2.1INTRODUCTION:**

In this chapter we will be describing database of the system tables inside that database and the way those tables were created in, addition on that in this chapter we will show table views from original tables, the relationship between created tables that are in this database.

In database of this system be ready to look at operation used on some table in this database so that you can know what to and what not to do on given entity.

Note that to develop this system database there are some material which were used so that we can get on final output of system as was described in chapter one “system analyses”. Among these the important one is xampp saver MySQL, so now let together navigate this database system.

**2.2 SECTION1:**

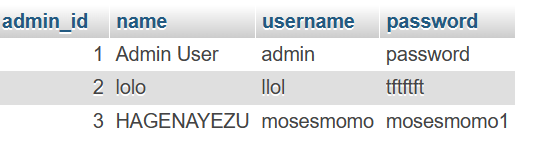
**2.2.1 ENTITIES**

1. Description of all entities and their corresponding attributes that are in this data base

#### admin table database structure

This is table is table that will be only created by admin and will hold other system users apart from admin it will give them username and password that they will use to login.

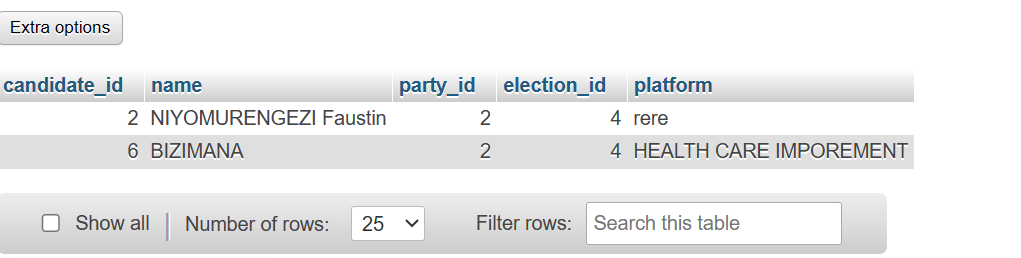
**Note:** the created admin will have ability and right in the system.



|  |
| --- |
| CREATE TABLE admins (  admin\_id INT NOT NULL AUTO\_INCREMENT,  name VARCHAR(255) NOT NULL,  username VARCHAR(255) NOT NULL,  password VARCHAR(255) NOT NULL,  PRIMARY KEY (admin\_id)  ) |

***candidates table database structure:***

This table to contain all candidates in the election process and where every candidate sign up here through filling their names, party that they are belong and platforms with also election type.



CREATE TABLE candidates (

candidate\_id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

party\_id INT NOT NULL,

election\_id INT NOT NULL,

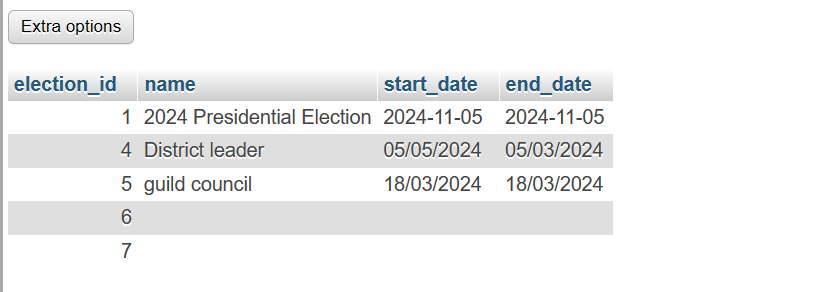
platform VARCHAR(255) NOT NULL,

PRIMARY KEY (candidate\_id),

);

#### Election table database structure

This table consists of the information related to the type/kind of election which is going to take a place. So as an admin of this system, you have to mention the election-type by demonstrating the name of it like presidential, district leader etc. also election information is characterized by starting date and ending ones so as to be well programmed and arranged.



CREATE TABLE elections (

election\_id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

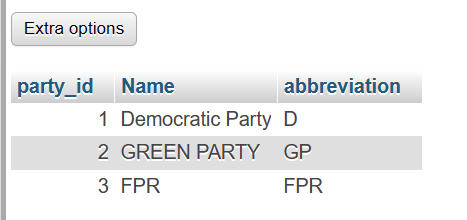
start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

);

#### Party table database structure

The party table in database stores informations about the parties that are participating in the election such as their names ,abbreviation and other relevant details. So this table help to organize ang manage different parties involved in voting process



CREATE TABLE parties (

party\_id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

abbreviation VARCHAR(255) NOT NULL,

PRIMARY KEY (party\_id)

);

#### Voter table database structure

This table in database of the online voting system stores the information about individual voters including their personal data like names, address, date of birth, email, username, password etc. here only admin can view all voters but only one vote can make CRUD operation on his/her account created in the system.



CREATE TABLE voters (

voter\_id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

date\_of\_birth DATE NOT NULL,

address VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL,

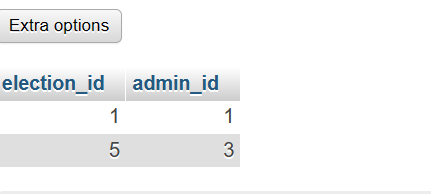
phone\_number VARCHAR(255) NOT NULL,

PRIMARY KEY (voter\_id)

);

#### Election admin table database structure

This tables consists of data of admins of different elections that are taking place on our system.



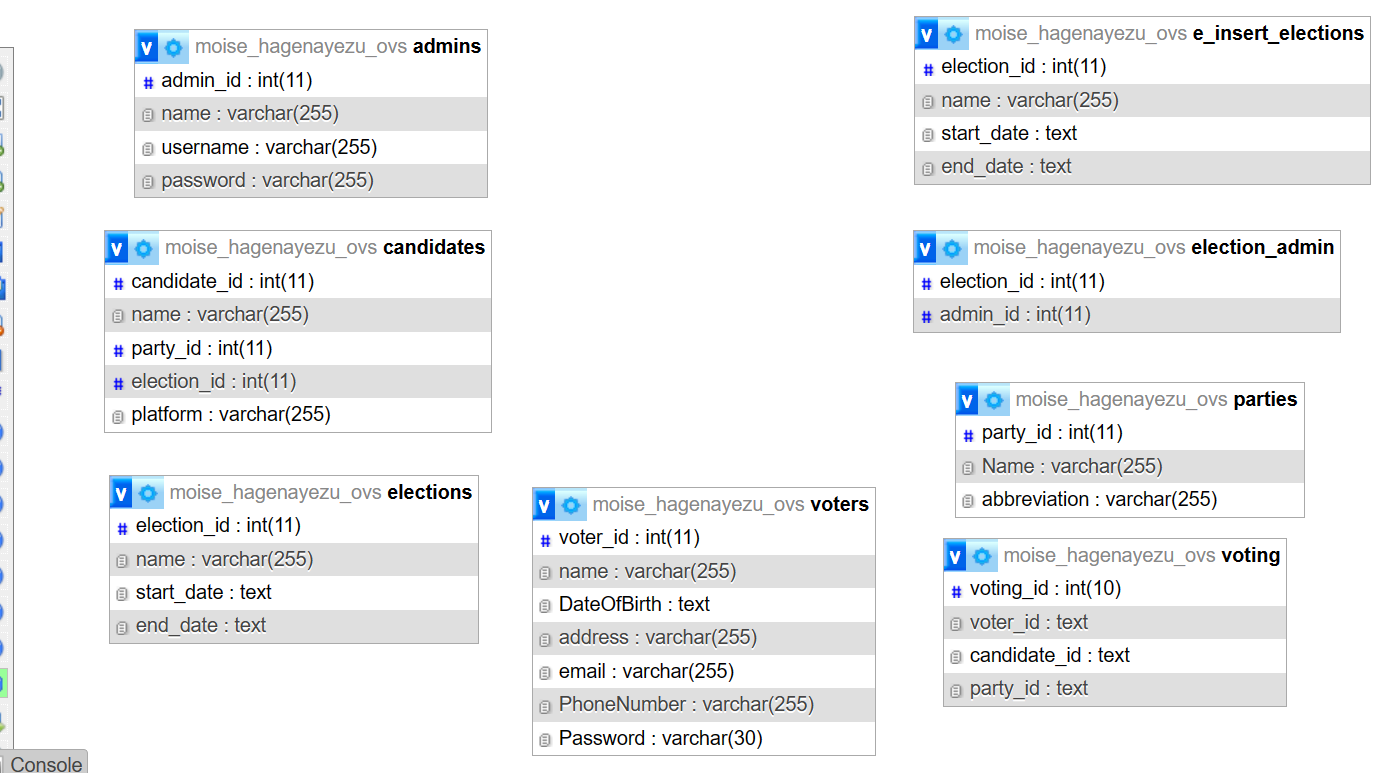
CREATE TABLE election\_admin (

election\_id INT NOT NULL, an identifier of chosen election

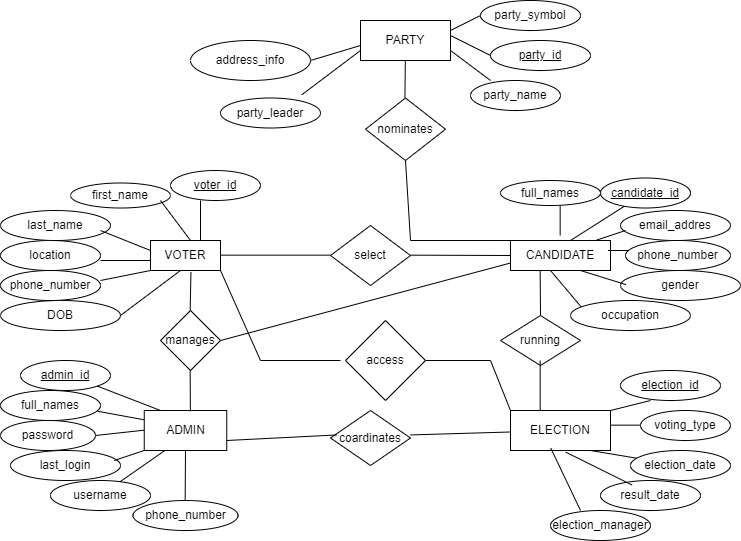
admin\_id INT NOT NULL, selected admin to control that election

2.2.2 LOGICAL DATABASE MODEL

The figure below describes how table related to one another logically in this system. What to say about this graph is that foreign key and primary one will generate these relationship as shown on this logical database model as well as entity relationship diagram below that.



**ENTITY RELATIONSHIP DIAGRAM**

****

**2.3 SECTION2:SQL**

In this sub unit we will be describing techniques especially SQL queries used to create, delete tables as well as inserting and deleting data in the tables in that database.

**1. CREATE DATABASE online\_voting\_system**

**2. Write queries to create all the tables and relationships of your system**

**CREATE TABLE voters (**

**voter\_id INT NOT NULL AUTO\_INCREMENT,**

**name VARCHAR(255) NOT NULL,**

**date\_of\_birth DATE NOT NULL,**

**address VARCHAR(255) NOT NULL,**

**email VARCHAR(255) NOT NULL,**

**phone\_number VARCHAR(255) NOT NULL,**

**PRIMARY KEY (voter\_id)**

**);**

**CREATE TABLE candidates (**

**candidate\_id INT NOT NULL AUTO\_INCREMENT,**

**name VARCHAR(255) NOT NULL,**

**party\_id INT NOT NULL,**

**election\_id INT NOT NULL,**

**platform VARCHAR(255) NOT NULL,**

**PRIMARY KEY (candidate\_id),**

**FOREIGN KEY (party\_id) REFERENCES parties(party\_id),**

**FOREIGN KEY (election\_id) REFERENCES elections(election\_id)**

**);**

**CREATE TABLE parties (**

**party\_id INT NOT NULL AUTO\_INCREMENT,**

**name VARCHAR(255) NOT NULL,**

**abbreviation VARCHAR(255) NOT NULL,**

**PRIMARY KEY (party\_id)**

**);**

**CREATE TABLE admins (**

**admin\_id INT NOT NULL AUTO\_INCREMENT,**

**name VARCHAR(255) NOT NULL,**

**username VARCHAR(255) NOT NULL,**

**password VARCHAR(255) NOT NULL,**

**PRIMARY KEY (admin\_id)**

**);**

**CREATE TABLE elections (**

**election\_id INT NOT NULL AUTO\_INCREMENT,**

**name VARCHAR(255) NOT NULL,**

**start\_date DATE NOT NULL,**

**end\_date DATE NOT NULL,**

**PRIMARY KEY (election\_id)**

**);**

**CREATE TABLE election\_admin (**

**election\_id INT NOT NULL,**

**admin\_id INT NOT NULL,**

**PRIMARY KEY (election\_id, admin\_id),**

**FOREIGN KEY (election\_id) REFERENCES elections(election\_id),**

**FOREIGN KEY (admin\_id) REFERENCES admins(admin\_id)**

**);**

**3. write queries to insert data into your tables.**

**-- Insert a new voter**

**INSERT INTO voters (name, date\_of\_birth, address, email, phone\_number)**

**VALUES ('John Doe', '1980-01-01', '123 Main Street', 'john.doe@example.com', '123-456-7890');**

**-- Insert a new candidate**

**INSERT INTO candidates (name, party\_id, election\_id, platform)**

**VALUES ('Jane Doe', 1, 2024, 'My platform is to improve education and healthcare.');**

**-- Insert a new party**

**INSERT INTO parties (name, abbreviation)**

**VALUES ('Democratic Party', 'D');**

**-- Insert a new admin**

**INSERT INTO admins (name, username, password)**

**VALUES ('Admin User', 'admin', 'password');**

**-- Insert a new election**

**INSERT INTO elections (name, start\_date, end\_date)**

**VALUES ('2024 Presidential Election', '2024-11-05', '2024-11-05');**

**-- Insert a relationship between an election and an admin**

**INSERT INTO election\_admin (election\_id, admin\_id)**

**VALUES (2024, 1);**

**4. Write queries to display all the information in your tables.**

**-- Display all voters**

**SELECT \* FROM voters;**

**-- Display all candidates**

**SELECT \* FROM candidates;**

**-- Display all parties**

**SELECT \* FROM parties;**

**-- Display all admins**

**SELECT \* FROM admins;**

**-- Display all elections**

**SELECT \* FROM elections;**

**-- Display all relationships between elections and admins**

**SELECT \* FROM election\_admin;**

**5. Write a query to update information in any of the two tables of your system.**

**UPDATE candidates**

**SET name = 'New Name'**

**WHERE candidate\_id = 1;**

**UPDATE voters**

**SET name = 'New Name'**

**WHERE voter\_id = 1;**

**Section III**

**1. Create a view to insert data into your tables.**

**CREATE VIEW v\_insert\_voters AS**

**SELECT name, date\_of\_birth, address, email, phone\_number INTO voters**

**FROM <source\_table>;**

**INSERT INTO v\_insert\_voters**

**VALUES ('John Doe', '1980-01-01', '123 Main Street', 'john.doe@example.com', '123-456-7890');**

**2. Create a view to display all the information in your tables.**

**CREATE VIEW v\_all\_info AS**

**SELECT \* FROM voters**

**UNION ALL**

**SELECT \* FROM candidates**

**UNION ALL**

**SELECT \* FROM parties**

**UNION ALL**

**SELECT \* FROM admins**

**UNION ALL**

**SELECT \* FROM elections**

**UNION ALL**

**SELECT \* FROM election\_admin;**

**3. Create a view to update information in any of the two tables of your system.**

**CREATE VIEW v\_update\_voter AS**

**SELECT voters.voter\_id, parties.party\_id**

**FROM voters**

**INNER JOIN parties ON voters.party\_id = parties.party\_id**

**WHERE voters.voter\_id = 1;**

**UPDATE v\_update\_voter**

**SET voters.name = 'New Name', voters.address = 'New Address', parties.name = 'New Party Name';**

**4. Create a view to delete data in any two of your tables according to any simple condition of your choice.**

**CREATE VIEW v\_delete\_voters AS**

**SELECT voters.voter\_id, elections.election\_id**

**FROM voters**

**INNER JOIN elections ON voters.election\_id = elections.election\_id**

**WHERE elections.election\_id = 2024 AND voters.state = 'CA';**

**DELETE FROM v\_delete\_voters;**

**5. In your database, create one view of your choice that considers sub-query.**

**CREATE VIEW v\_oldest\_voters AS**

**SELECT voters.name, voters.date\_of\_birth**

**FROM voters**

**WHERE voters.date\_of\_birth <= (SELECT MIN(voters.date\_of\_birth) FROM voters);**

**Section IV**

**1. Create a stored procedure to insert data into your tables.**

**CREATE PROCEDURE sp\_insert\_voter (**

**IN name VARCHAR(255),**

**IN date\_of\_birth DATE,**

**IN address VARCHAR(255),**

**IN email VARCHAR(255),**

**IN phone\_number VARCHAR(255)**

**)**

**BEGIN**

**INSERT INTO voters (name, date\_of\_birth, address, email, phone\_number)**

**VALUES (name, date\_of\_birth, address, email, phone\_number);**

**END;**

**CREATE PROCEDURE sp\_insert\_candidate (**

**IN name VARCHAR(255),**

**IN party\_id INT,**

**IN election\_id INT,**

**IN platform VARCHAR(255)**

**)**

**BEGIN**

**INSERT INTO candidates (name, party\_id, election\_id, platform)**

**VALUES (name, party\_id, election\_id, platform);**

**END;**

**2. Create a stored procedure to display all the information in your tables.**

**CREATE PROCEDURE sp\_display\_all\_info ()**

**BEGIN**

**SELECT \* FROM voters**

**UNION ALL**

**SELECT \* FROM candidates**

**UNION ALL**

**SELECT \* FROM parties**

**UNION ALL**

**SELECT \* FROM admins**

**UNION ALL**

**SELECT \* FROM elections**

**UNION ALL**

**SELECT \* FROM election\_admin;**

**END;**

**3. Create a stored procedure to update information in any of the two tables of your system.**

**CREATE PROCEDURE sp\_update\_data (**

**IN table\_name VARCHAR(255),**

**IN column\_name VARCHAR(255),**

**IN new\_value VARCHAR(255),**

**IN where\_clause VARCHAR(255)**

**)**

**BEGIN**

**UPDATE table\_name**

**SET column\_name = new\_value**

**WHERE where\_clause;**

**END;**

**CALL sp\_update\_data ('voters', 'name', 'New Name', 'voter\_id = 1');**

**CALL sp\_update\_data ('voters', 'name', 'New Name', 'voter\_id = 1');**

**CALL sp\_update\_data ('voters', 'address', 'New Address', 'voter\_id = 1');**

**CALL sp\_update\_data ('candidates', 'name', 'New Name', 'candidate\_id = 1');**

**CALL sp\_update\_data ('parties', 'name', 'New Party Name', 'party\_id = (SELECT party\_id FROM candidates WHERE candidate\_id = 1)');**

**4. Create a stored procedure to delete data in any two of your tables according to any simple condition of your choice.**

**CREATE PROCEDURE sp\_delete\_data (**

**IN table1\_name VARCHAR(255),**

**IN table2\_name VARCHAR(255),**

**IN column1 VARCHAR(255),**

**IN column2 VARCHAR(255),**

**IN condition VARCHAR(255)**

**)**

**BEGIN**

**DELETE FROM table1\_name**

**INNER JOIN table2\_name ON table1\_name.column1 = table2\_name.column2**

**WHERE condition;**

**END;**

CALL sp\_delete\_data ('voters', 'elections', 'voter\_id', 'election\_id', 'election\_id = 2024 AND voters.state = "CA"');

CALL sp\_delete\_data ('voters', 'elections', 'voter\_id', 'election\_id', 'election\_id = 2024');

CALL sp\_delete\_data ('candidates', 'elections', 'candidate\_id', 'election\_id', 'election\_id = 2024');

CALL sp\_delete\_data ('parties', 'candidates', 'party\_id', 'candidate\_id', 'candidate\_id IN (SELECT candidate\_id FROM candidates WHERE election\_id = 2024)');

**5. In your database, stored the procedure view of your choice that considers subquery**

CREATE PROCEDURE sp\_get\_oldest\_voters ()

BEGIN

RETURN (SELECT \* FROM voters

WHERE date\_of\_birth <= (SELECT MIN(date\_of\_birth) FROM voters));

END;

SELECT \* FROM sp\_get\_oldest\_voters()

INNER JOIN elections ON sp\_get\_oldest\_voters().election\_id = elections.election\_id

WHERE elections.election\_id = 2024;

**Section V**

**1. Create after inserting triggers for any two tables of your choice.**

CREATE TRIGGER trg\_insert\_voter\_into\_log AFTER INSERT ON voters

FOR EACH ROW

INSERT INTO vote\_log (voter\_id, operation, timestamp)

VALUES (NEW.voter\_id, 'INSERT', GETDATE());

CREATE TRIGGER trg\_insert\_candidate\_into\_log AFTER INSERT ON candidates

FOR EACH ROW

INSERT INTO vote\_log (candidate\_id, operation, timestamp)

VALUES (NEW.candidate\_id, 'INSERT', GETDATE());

INSERT INTO voters (name, date\_of\_birth, address, email, phone\_number)

VALUES ('John Doe', '1980-01-01', '123 Main Street', 'john.doe@example.com', '123-456-7890');

INSERT INTO candidates (name, party\_id, election\_id, platform)

VALUES ('Jane Doe', 1, 2024, 'Make America Great Again');

**2. Create after-update triggers for any two tables of your choice.**

**CREATE TRIGGER trg\_update\_voter\_in\_log AFTER UPDATE ON voters FOR EACH ROW**

BEGIN

INSERT INTO vote\_log (voter\_id, operation, timestamp)

VALUES (NEW.voter\_id, 'UPDATE', GETDATE());

END;

**CREATE TRIGGER trg\_update\_candidate\_in\_log AFTER UPDATE ON candidates FOR EACH ROW**

BEGIN

INSERT INTO vote\_log (candidate\_id, operation, timestamp)

VALUES (NEW.candidate\_id, 'UPDATE', GETDATE());

END;

**3. Create after deleting triggers for any two tables of your choice.**

**CREATE TRIGGER trg\_delete\_voter\_from\_log AFTER DELETE ON voters FOR EACH ROW**

BEGIN

INSERT INTO vote\_log (voter\_id, operation, timestamp)

VALUES (OLD.voter\_id, 'DELETE', GETDATE());

END;

**CREATE TRIGGER trg\_delete\_candidate\_from\_log AFTER DELETE ON candidates FOR EACH ROW**

BEGIN

INSERT INTO vote\_log (candidate\_id, operation, timestamp)

VALUES (OLD.candidate\_id, 'DELETE', GETDATE());

END;

**Section VI**

**1. Create a user with your name as username and your student number as password and grant all privileges to the created user.**

mysql -u root -p

create user 'hagenayezumoise'@'localhost'IDENTIFIED BY '222006319';

GRANT ALL PRIVILEGES ON \*.\* TO 'hagenayezumoise'@'localhost';

FLUSH PRIVILEGES;

exit;

**2. Create a user with your "names\_semi" as username and your student number as password and give him insert, update, and delete privileges to the created user.**

**mysql -u root -p**

**CREATE USER 'moise'@'localhost' IDENTIFIED BY ‘222006319';**

**GRANT INSERT, UPDATE, DELETE ON \*.\* TO 'moise'@'localhost';**

**FLUSH PRIVILEGES;**

**exit;**

**3. Revoke insert privileges to the last user you created.**

mysql -u root -p

REVOKE INSERT ON \*.\* FROM 'moise'@'localhost';

FLUSH PRIVILEGES;

exit;

**Conclusion:**

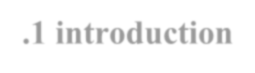
By concluding on this database we have achieved more on this system using MySQL and the system in terms of keeping data is working well with different operation we have settled on it in some tables like revoking some activities, triggers are created in this database and relationship of much of these tables in this database are there to link them one by one. So this will lead us to creating the link between data store which is database and the final user of the system which will be focusing more on user interface and connectivity.

**Reference:**

1. **Six-Step Relational Database Design Second Edition (Fidel A Captain, 2016)**
2. [**www.erbessd-instruments.com**](http://www.erbessd-instruments.com/)



**3**



**.1**

**introduction**

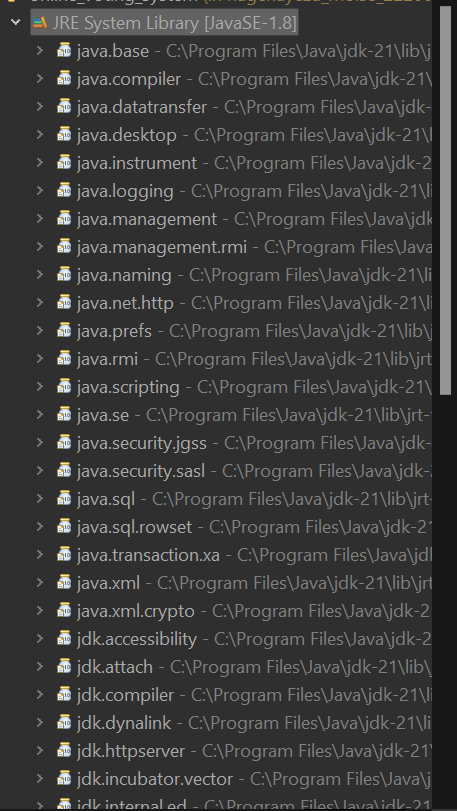


In this chapter I will be describing how powerful general purpose **programming** language was used to create the analyzed system. Under this chapter I will undergo full detail of how everything will function together with database that have been describe above and how it cope with full analyzed system.

**3.2 Tools used to develop this system in java programming:**

Eclipse IDE: an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development, and, until 2016, was the most popular.

**JAR stands for Java Archive**. It's a file format based on the popular ZIP file format and is used for aggregating many files into one. Although JAR can be used as a general archiving tool, the primary motivation for its development was so that Java applets and their requisite components.



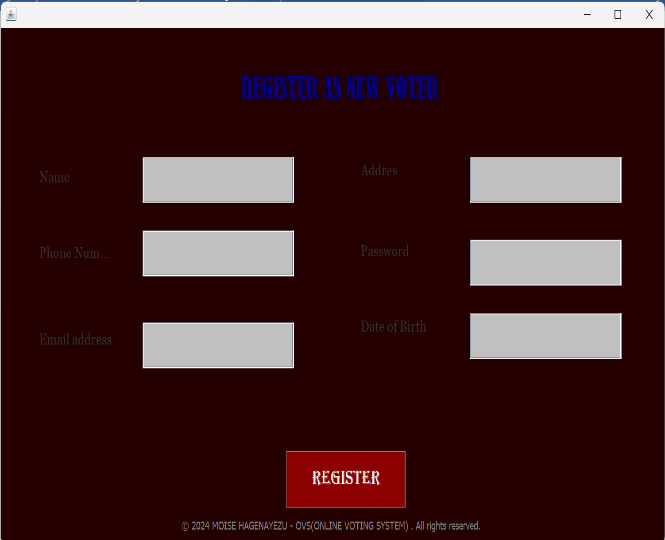
**MySQL Connectors**. **MySQL** provides standards-based drivers for JDBC, ODBC, and .Net enabling developers to build database applications in their language

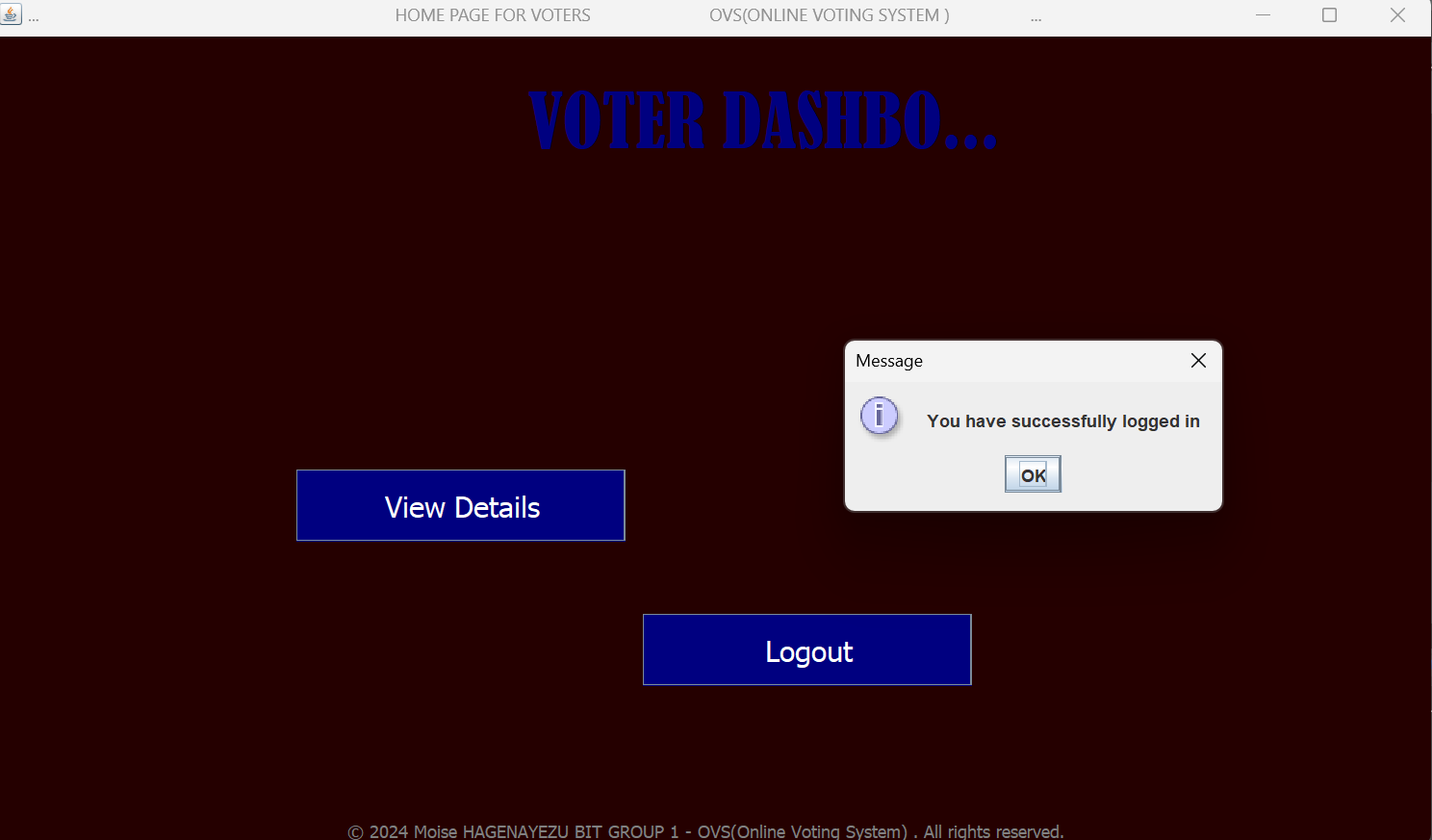
### 3.3 Forms description

So let look together how the system will function one by one from the beginning up to the end.

1. **Registration of new voter**

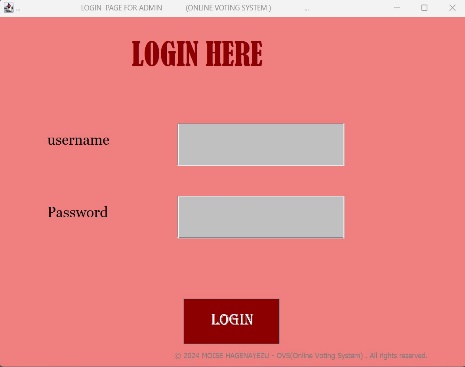
Here someone who is eligible to vote, he/she use this form to fill information related to him/her and then store those information into MySQL database through GUI. This form provide a user interface for voters for being registered in this online voting system. After logging in It direct u user to his/her dashboard as you se below;

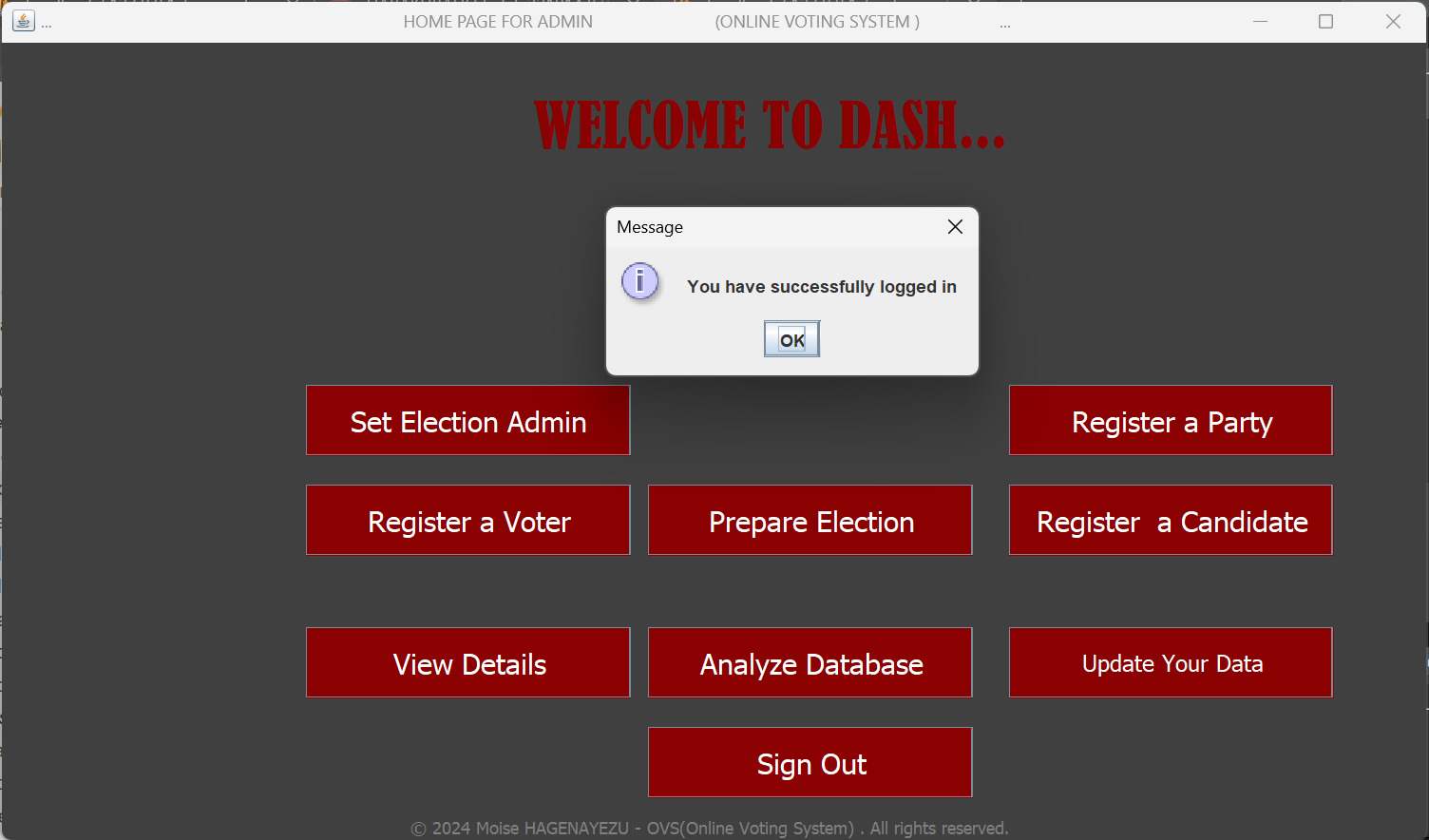
****



1. **Login form for admin**

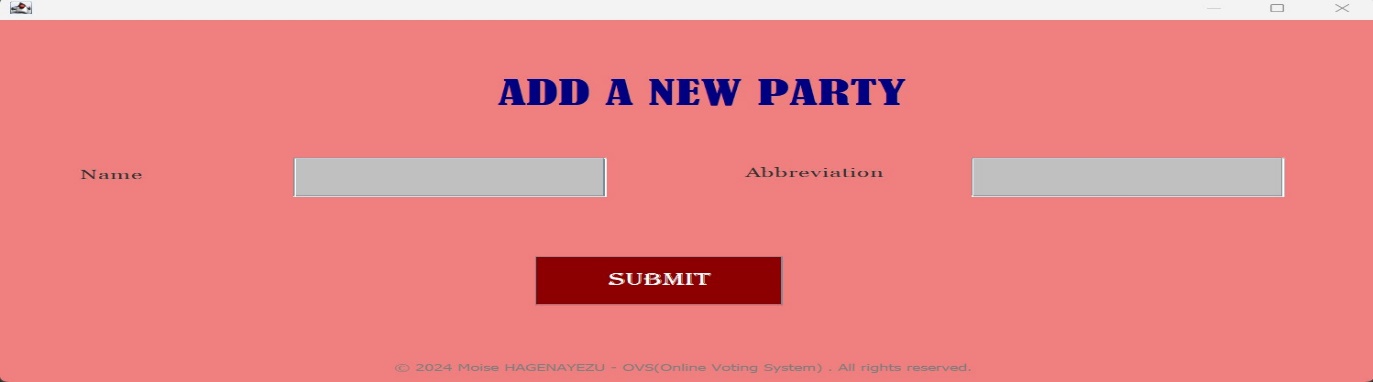
This form is used to help those admins of this system to log in after creating their accounts in our database. Each one has own username and password. After login through this form it direct you to the dashboard where you can get graphic user interface related to your information saved. He/she can make operation of CRUD(create, read, update and delete)





1. **Party form**

This form is accessed by an admin of the system where he/she add or introduce new party to our system so as to let the voters get information about it.



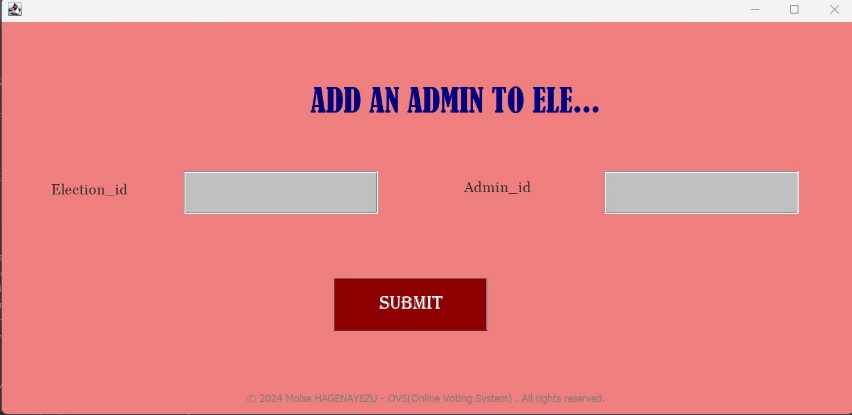
1. **Admin form**

This is concerned with the registration of an admin in this system so as to get access to the election which is taking place. Here the admin has all right to access all system information where he can read, delete, update, and so on. After register, he/she can go to login and fill username and password that created and then get dashboard after login successful.



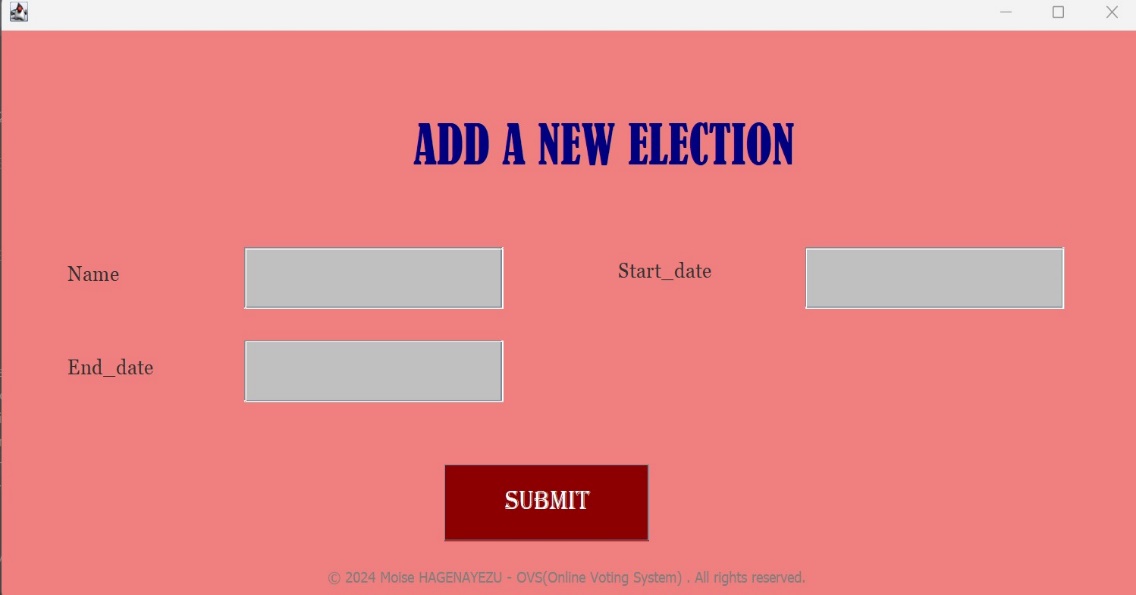
1. **Addition of admin to election**

as we know, admin has all right to do CRUD operations in this system. So we can add many admins to work together with the existed ones through entering his identifier and the election identifier he/she want to work with. But first of all this admin have to be registered and be existed to the system.

****

1. **Adding election**

This form consists of data for admins of different elections that are taking place on our system. It can be accessed by only admin. An admin set the new election starting date and ending date as generally we know.

****

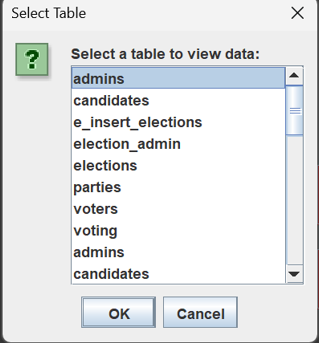
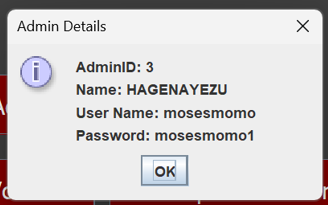
1. **Candidate form**

This form help tin registration of all candidates in the election process and where every candidate sign up here through filling their names, party that they are belong and platforms with also election type and its identifier stored in the database.



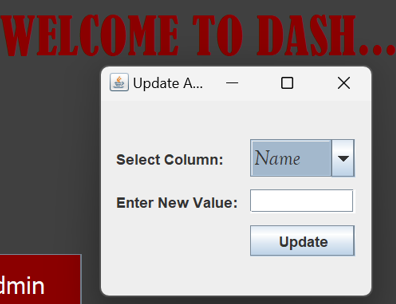
* 1. **view your data**

Here admin can view data that are inserted by him/her as shown below and view tables details as shown below;



**3.5Updating your data**

Here admin can update the tables name in the way she want after getting to the dashboard.



**3.6 Special buttons:**

**Ok button:**

This is button where pressing on it bring you an ongoing permission or show you that you perform well.

**Sign out button**:

At this button you will get out of the system so that you will be required to enter login

Creditantial to enter again.

**3.7 Conclusion:**

By concluding this chapter concerns with java programming especially in my developed

System of online voting, we can say that I have final product that I was expecting to have it, the

Manipulation of data is going well the design is there with special appearance, but there

Much to go on and that need to be improved will be gained from external view apart from

System developer. The use of java and JDBC technology enhances the portability and scalability of the application

**Reference:**

1. Java Performance: The Definite Guide First Edition (Scott Oaks, 2017)
2. www.eclipse.org>windowbuilider
3. www.vogella.com>article
4. www.formdev.com>ides>eclipse