

ONLINE VOTING SYSTEM

BY Sahil Bengloor

18057

Table of Contents

1 Introduction

2 Purpose

3 Scope

3.1 product prespective

3.2 Function

3.3 User Characteristic

3.4 Constraints

3.5 Assumptions and Dependencies

4 Product Perspective

4.1 Product Function

4.2 User Classes and Characteristics

4.3 Design and Implementation Constraints

4.4 Assumptions and Dependencies

5 Overview

6 Overall Description

6.1 Product perspective

6.2 Production Function

6.3 User Characteristics

6.4 General Constraints

6.5 Assumption and Dependencies

7 External Interface Requirments

7.1 User Interfaces

7.2 Hardware Interface

7.3 Software Interface

7.4 Communication Interface

8 Functional Requirments

8.1 User login

8.2 Candidate Selection

8.3 Voter verification and validation

8.4 Result Calculation and Display

8.5 Administrator Management and Monitoring

9 Performance Requirments

9.1 Response Time

9.2 Scalability

9.3 Availability

9.4 Security

9.5 Backup and Recovery

10 Design Constraints

11 Feasibility Study

11.1 Project Description

11.2 Technical Feasibility

11.3 Market Feasibility

11.4 Financial Feasibility

11.5 Legal and Ethical Feasibility

11.6 Operational Feasibility

11.7 Risk Assessment

12 Timeline Chart

13 Component Diagram

14 Case Diagram

15 Class Diagram

16 Detailed Diagram

17 Sequence Diagram

18 State Diagram

19 Different Models

19.1 Incremental Model

19.2 Waterfall Model

19.3 Spiral Model

19.4 Agile Model

19.5 V-Model

20 Future Scope

21 Conclusion

22 References

1. Introduction

The Online Voting System is a simple command-line application that simulates an electronic voting process. The system prompts the user to enter their name and phone number, and then allows them to cast their vote by selecting one of the available options: FyBsc IT, SyBsc IT, or TyBsc IT. After the user has submitted their vote, they can check the results to see the current tally. The system keeps track of the number of votes for each option and displays this information when the user checks the results. The system also indicates whether the user has already voted or not. This project can be used as a starting point for developing more sophisticated online voting systems that can handle larger scale elections and provide additional features such as voter authentication and result verification.

2. Purpose

The purpose of the Online Voting System is to provide a simple and user-friendly interface for casting and counting votes in a simulated election. The system is designed to demonstrate the basic principles of electronic voting and to serve as a starting point for developing more advanced voting systems. The system allows users to enter their name and phone number, cast their vote by selecting one of the available options, and check the results to see the current tally. By providing a convenient and accessible way to vote, the system aims to encourage participation in the democratic process and promote transparency and accountability in election outcomes. The Online Voting System can be used for educational purposes, to simulate elections in small organizations or groups, or as a basis for further development and customization to meet specific voting needs.

3.Scope

The scope of the project defines what the software system will encompass. It sets the boundaries by specifying what functionalities and features are included and what falls outside the project's purview. Consider the following components:

3.1 Product Perspective

- Describe how your software system fits into the larger context.
- Identify any external systems it interacts with (e.g., databases, APIs, other applications).
- Highlight the interfaces and dependencies.

3.2 Functions

- Enumerate the primary functions or features your software will provide.
- Be specific about what actions users can perform (e.g., user registration, voting, result display).
- Avoid vague statements; instead, provide clear descriptions.

3.3 User Characteristics

- Define the intended users (e.g., voters, administrators, candidates).
- Specify their roles, responsibilities, and expectations.
- Consider user demographics, technical expertise, and any special requirements.

3.4 Constraints

- Identify any limitations or restrictions affecting the project.
- Examples include time constraints, budget limitations, hardware/software compatibility, and legal/regulatory requirements.

3.5 Assumptions and Dependencies

- List any assumptions made during project planning.
- Document external factors that impact the project (e.g., availability of third-party APIs, data sources).

4. Product Perspective

The software product is a standalone system and not apart of a larger system. The system will be made up of two parts, one running visible directly to the administrator on the server machine and the other visible to the end users, in this case the voters, through web pages. The two users of the system, namely the voters and the admin interact with the system in different ways. The admin configures the whole system according to it's needs on the server where the system is running. The voters cast their votes using the web interface provided. These votes are accepted by the system on the server.

4.1 Product Functions

On the Admin side, the system can be used to create/update/delete the election details (posts, candidates, electoral rolls etc). The Admin should be able to specify the different attributes it wants for Software Requirements Specification for Online Voting posts/candidates of a particular election instance and voters. From the voters perspective, the system is used to help them cast their votes and after the elections are over, allow them to view the results, which are automatically posted on the same site after the election duration is over.

4.2 User Classes and Characteristics

The users can be divided into two main classes:

- The Admin: They manage the entire Voting System Software and Conduct the Elections. They act as the Election Authority.

- The Voters: The voters should have a basic knowledge of how to use a web browser and navigate through web pages. The voters should be aware that they have to keep their user-id and password confidential.

4.3 Design and Implementation Constraints

- GUI is only in English.
- This system is working for in a Simple GUI
- Simple Java Program Using Different Packages
- User should have basic knowledge of computer. Software Requirements Specification for Online Voting

4.4 Assumptions and Dependencies

- The end user should have a basic knowledge of English and computer usage.
- Administrator is created in the system already.
- The voting results will be managed and calculated by the admin.
- Roles and tasks are predefined

5. Overview

The Online Voting System is a Java-based application that enables users to cast their votes securely and efficiently. The system provides a user-friendly interface that allows users to register, log in, and cast their votes for their preferred candidates. The system also provides an administrator interface to manage the voting process, view voter statistics, and declare the results.

The Online Voting System aims to provide a secure, transparent, and accessible voting platform that eliminates the need for physical voting and reduces the risk of fraud and errors. The system ensures that each voter can only cast one vote, and the results are accurately calculated and displayed.

The system's main features include:

- User registration and login
- Candidate selection and voting

- Voter verification and validation
- Result calculation and display
- Administrator management and monitoring

The Online Voting System is designed to be user-friendly, secure, and scalable, making it an ideal solution for various voting scenarios, such as elections, surveys, and polls.

6. Overall Description

6.1 Product Perspective:

The Online Voting System is a Java-based application that enables users to cast their votes securely and efficiently. The system provides a user-friendly interface that allows users to register, log in, and cast their votes for their preferred candidates. The system also provides an administrator interface to manage the voting process, view voter statistics, and declare the results.

6.2 Product Functions:

The Online Voting System provides the following functions:

- User registration and login
- Candidate selection and voting
- Voter verification and validation
- Result calculation and display
- Administrator management and monitoring

6.3 User Characteristics:

The Online Voting System is designed for the following users:

- Voters who want to cast their votes securely and efficiently

- Administrators who want to manage the voting process, view voter statistics, and declare the results

6.4 General Constraints:

The Online Voting System is subject to the following constraints:

- The system must comply with all relevant laws and regulations related to voting and data privacy.
- The system must be able to handle a large number of concurrent users without performance degradation.
- The system must be able to provide high availability and reliability.

6.5 Assumptions and Dependencies:

The Online Voting System makes the following assumptions and dependencies:

- Users have access to a device with an internet connection and a web browser.
- Users have a valid email address for registration and verification.
- The system assumes that the administrator is responsible for adding and managing the candidates and the voting process.
- The system assumes that the network infrastructure is secure and reliable.
- The system assumes that the database and server infrastructure can handle the load and provide high availability and reliability.

7. External Interface Requirements:

The Online Voting System has the following external interface requirements:

7.1 User Interfaces:

The system shall provide a user-friendly interface that allows users to register, log in, and cast their votes for their preferred candidates. The user interface shall be accessible through a web browser and shall be compatible with various devices, including desktops, laptops, tablets, and smartphones. The user interface shall be designed to be intuitive, easy to use, and accessible to users with disabilities.

7.2 Hardware Interface:

The system shall be compatible with various hardware platforms, including Windows, macOS, and Linux. The system shall support the following hardware interfaces:

- Keyboard and mouse input
- Monitor output
- Network connectivity

7.3 Software Interface:

The system shall support the following software interfaces:

- Web browsers: The system shall support the latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari.
- Operating systems: The system shall support the latest versions of Windows, macOS, and Linux.

7.4 Communication Interface:

The system shall support secure communication protocols, including HTTPS and SSL/TLS, to ensure the confidentiality, integrity, and authenticity of the data

transmitted between the client and the server. The system shall also support VPN and proxy servers to provide additional security and privacy

8. Functional Requirements:

The Online Voting System shall provide the following functional requirements:

8.1 User Registration and Login:

The system shall allow users to register and create an account using their email address and a secure password. The system shall also allow users to log in to their account using their email address and password.

8.2 Candidate Selection and Voting:

The system shall allow users to view the list of candidates and their details. The system shall allow users to select their preferred candidate and cast their vote. The system shall ensure that each user can only cast one vote.

8.3 Voter Verification and Validation:

The system shall verify and validate the voter's identity using various methods, including email verification, SMS verification, or government-issued ID verification. The system shall ensure that only eligible voters can cast their votes.

8.4 Result Calculation and Display:

The system shall calculate and display the results of the voting in real-time. The system shall provide various statistics, including the number of votes cast, the number of eligible voters, and the percentage of votes received by each candidate.

8.5 Administrator Management and Monitoring:

The system shall allow administrators to manage the voting process, including adding and removing candidates, managing the voter list, and monitoring the voting progress. The system shall provide various reports and analytics to help administrators make informed decisions

9. Performance Requirements:

The Online Voting System shall meet the following performance requirements:

9.1 Response Time:

The system shall respond to user requests within 2 seconds or less, including page load times, form submissions, and data processing.

9.2 Scalability:

The system shall be able to handle a large number of concurrent users without degrading the performance or availability of the system. The system shall be designed to scale horizontally and vertically to meet the changing demands of the users.

9.3 Availability:

The system shall be available 99.9% of the time, excluding scheduled maintenance windows. The system shall provide high availability through redundancy, failover, and disaster recovery mechanisms.

9.4 Security:

The system shall provide robust security features, including encryption, authentication, and authorization, to protect the confidentiality, integrity, and availability of the system and the data. The system shall comply with various security standards, including OWASP Top 10 and PCI DSS.

9.5 Backup and Recovery:

The system shall provide regular backups of the data and the system configuration. The system shall provide disaster recovery mechanisms to restore the system and the data in case of a failure or a disaster

10. Design Constraints:

The Online Voting System shall be designed within the following constraints:

- The system shall be developed using Java and related technologies. This constraint is necessary because the development team has expertise in Java and related technologies, and the development timeline and budget do not allow for learning and adopting new technologies.
- The system shall be deployed on a cloud-based infrastructure. This constraint is necessary to provide scalability, availability, and reliability, and to reduce the cost and complexity of managing the infrastructure.
- The system shall comply with various laws and regulations related to voting and data privacy. This constraint is necessary to ensure the legality and ethicality of the system and to protect the rights and interests of the users and the stakeholders.

11. Feasibility Study

11.1 Project Description:

The proposed project aims to develop an online voting system using Java, catering to the increasing demand for efficient and accessible voting mechanisms. This system will allow users to cast their votes securely and conveniently over the internet, eliminating the need for physical presence at polling stations. The primary objectives include enhancing the democratic process, increasing voter participation, and streamlining the election administration process.

11.2 Technical Feasibility:

- **Requirements and Constraints:**
 - Development in Java programming language.
 - Integration with web technologies such as the user interface.
 - Utilization of database systems like MySQL or PostgreSQL for data storage.
 - Implementation of secure authentication and encryption mechanisms.
- **Availability:**
 - Java development tools and frameworks are widely available and well-supported.
 - Abundance of resources and documentation for Java-based web development.
- **Scalability, Performance, and Security:**
 - Scalability will be ensured through efficient database design and optimization.
 - Performance will be monitored and improved through testing and optimization techniques.

- Security measures, including encryption, secure communication protocols, and access control, will be implemented to safeguard against unauthorized access and tampering.

11.3 Market Feasibility:

- **Demand:**
 - Growing demand for online voting systems due to their convenience and accessibility.
 - Potential users include governments, institutions, organizations, and communities seeking efficient election solutions.
- **Competition:**
 - Analysis of existing online voting systems and their features.
 - Identification of gaps or areas for improvement to differentiate the proposed system.
- **Niche Identification:**
 - Exploration of specialized markets or sectors where the proposed system can address specific needs or requirements.

11.4 Financial Feasibility:

- **Cost Estimation:**
 - Initial development costs, including hardware, software licenses, and personnel expenses.
 - Ongoing operational costs, such as hosting fees, maintenance, and support.
- **Revenue Projection:**
 - Potential revenue streams, such as licensing fees, subscription models, or service charges.
 - Assessment of the financial viability based on projected costs and revenues.

- **Cost-Benefit Analysis:**

- Evaluation of the expected benefits against the costs to determine the project's financial feasibility.

11.5 Legal and Ethical Feasibility:

- **Regulatory Compliance:**

- Identification of relevant laws and regulations governing elections, data privacy, and security.
- Ensuring compliance with legal requirements regarding voter authentication, data protection, and transparency.

- **Ethical Considerations:**

- Adherence to ethical principles, including fairness, integrity, and respect for voter privacy.
- Implementation of measures to prevent fraud, manipulation, and coercion.

11.6 Operational Feasibility:

- **Usability:**

- User-friendly interface design to facilitate ease of use for both administrators and voters.
- Accessibility features to accommodate users with disabilities or special needs.

- **Scalability and Reliability:**

- System architecture designed to handle peak loads during elections without compromising performance.
- Measures to ensure continuous availability and reliability, including backup and disaster recovery plans.

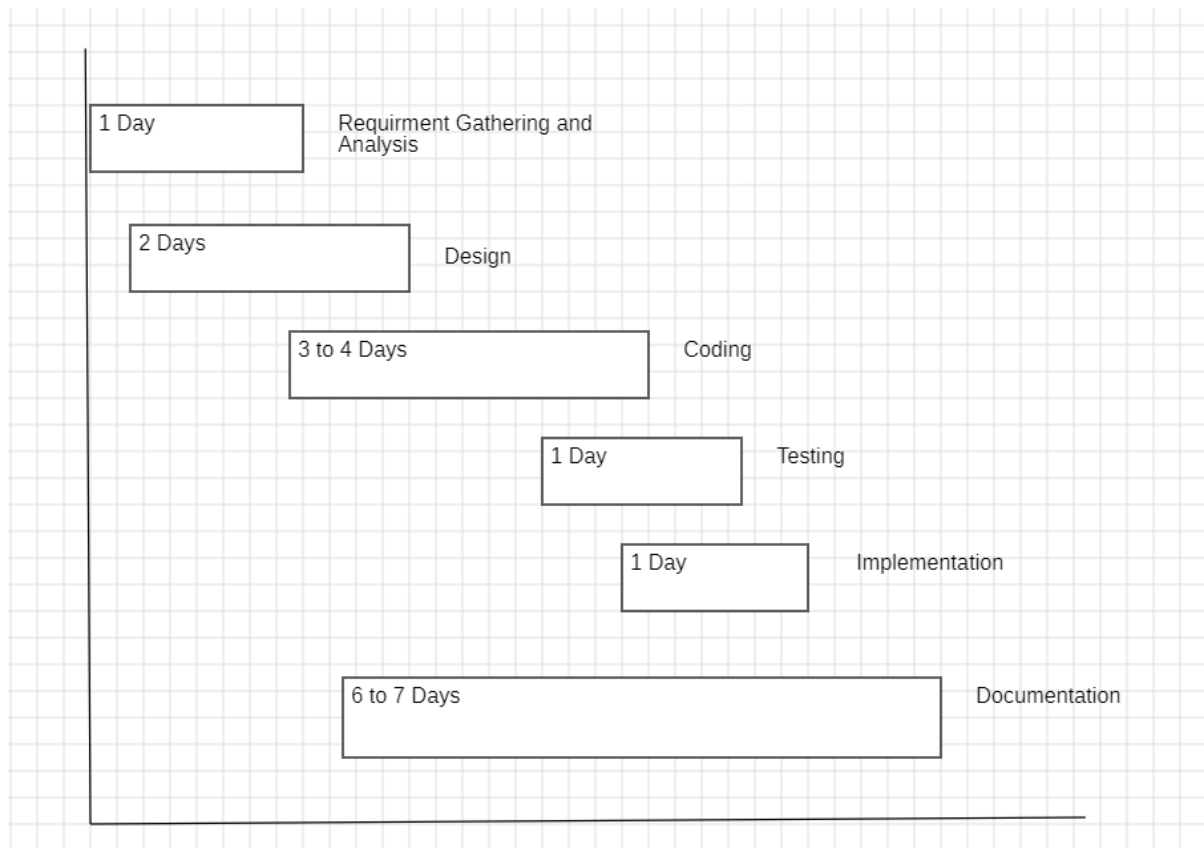
- **Risk Management:**

- Identification of operational risks, such as technical failures, cyber attacks, or system downtime.
- Development of risk mitigation strategies to minimize the impact of potential disruptions.

11.7 Risk Assessment:

- **Risk Identification:**
 - Identification of potential risks and uncertainties that could affect project success.
 - Risks include technical challenges, security vulnerabilities, regulatory changes, and operational disruptions.
- **Risk Analysis:**
 - Assessment of the likelihood and potential impact of each identified risk.
 - Prioritization of risks based on their severity and likelihood of occurrence.
- **Risk Mitigation:**
 - Development of risk mitigation plans to address and mitigate identified risks.
 - Implementation of preventive measures, contingency plans, and monitoring mechanisms.

12. TimeLine Chart



1. Requirement Gathering and Analysis: 1 Day + 2 Days

- Day 1: Initial meeting and understanding the project requirements.
- Day 2: Further analysis of the project requirements and defining the scope.

2. Design: 3 to 4 Days

- Day 3-4: Designing the online voting system, including system architecture and user interface.

3. Coding: 1 Day + 1 Day

- Day 1: Start coding the system.
- Day 2: Continue coding and complete the basic structure of the system.

4. Testing: 1 Day

- Day 1: Test the system for any bugs or issues.

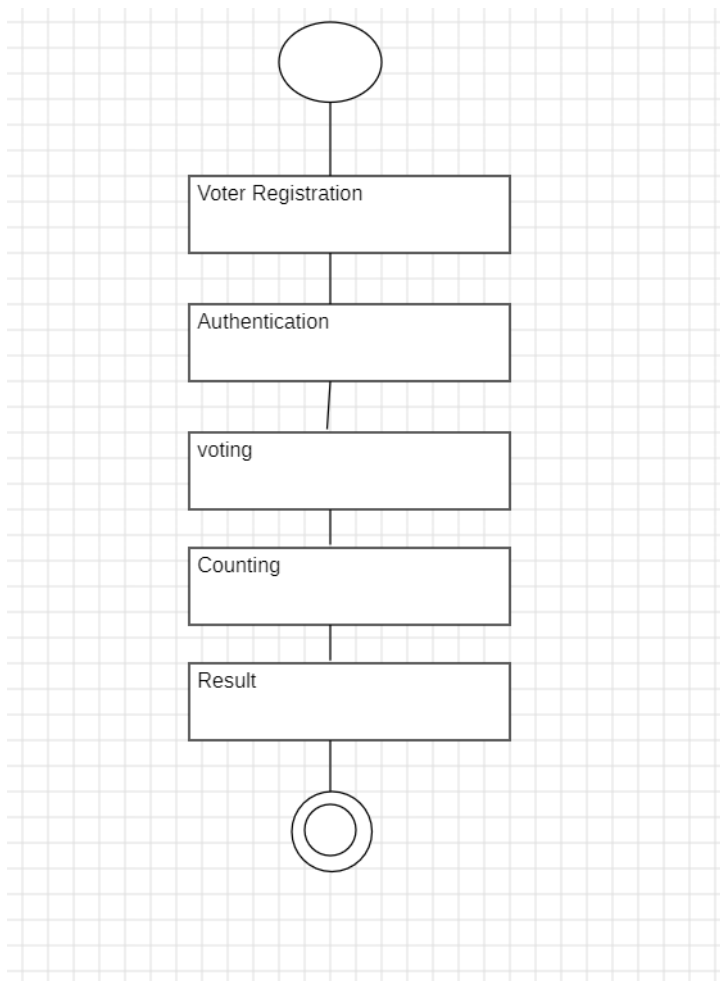
5. Implementation: Not specified in the timeline

- Details about the implementation are not provided in the timeline.

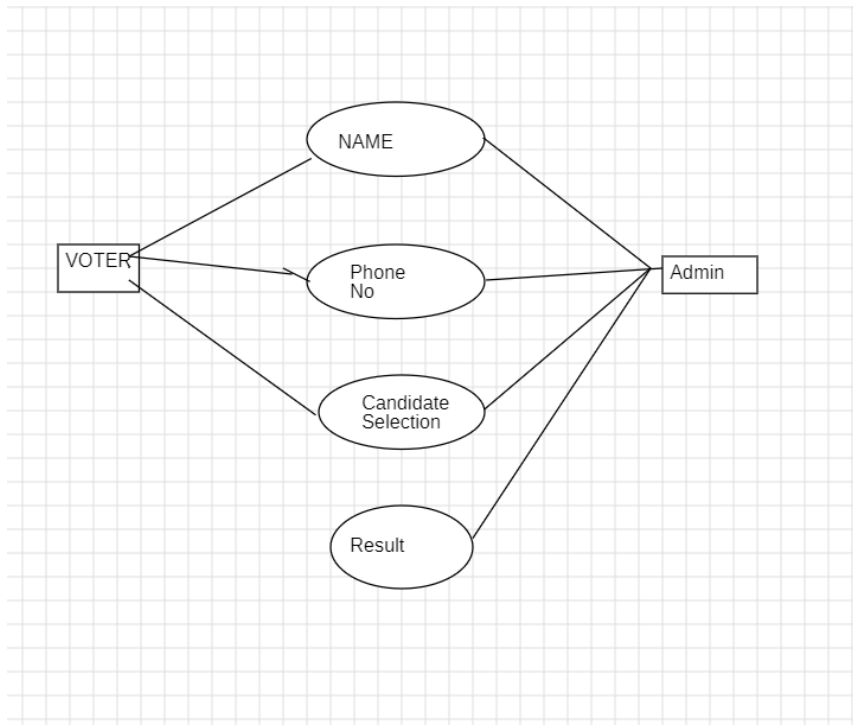
6. Documentation: 1 Day + 2 Days

- Day 1: Document the system requirements, design, and coding.
- Day 2: Continue with the documentation, including testing and implementation details.

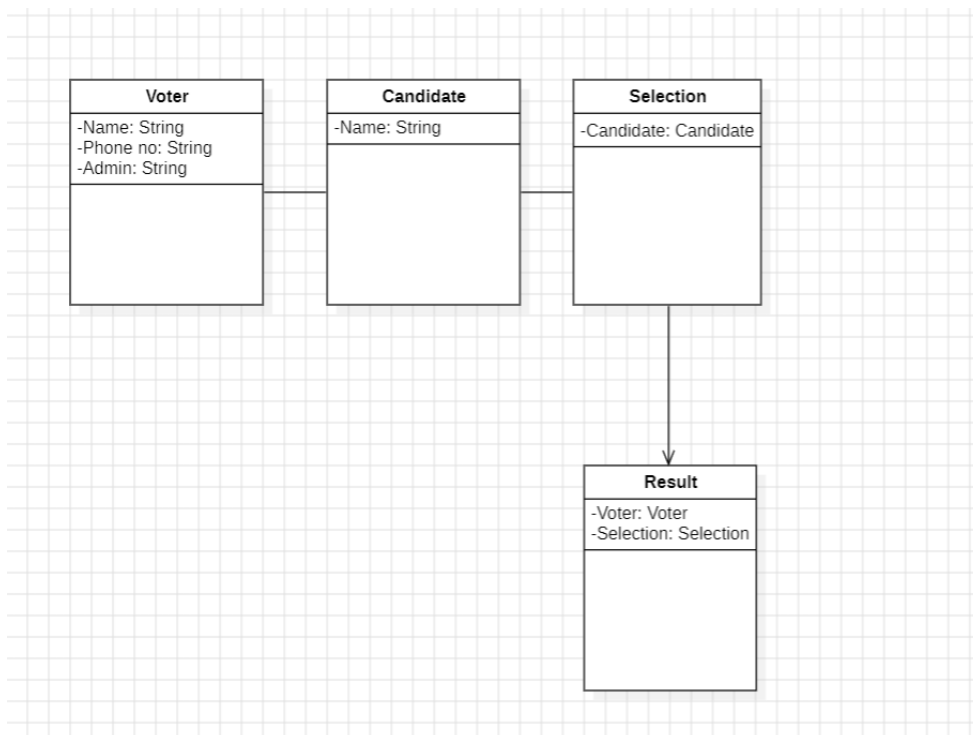
13 Component Diagram



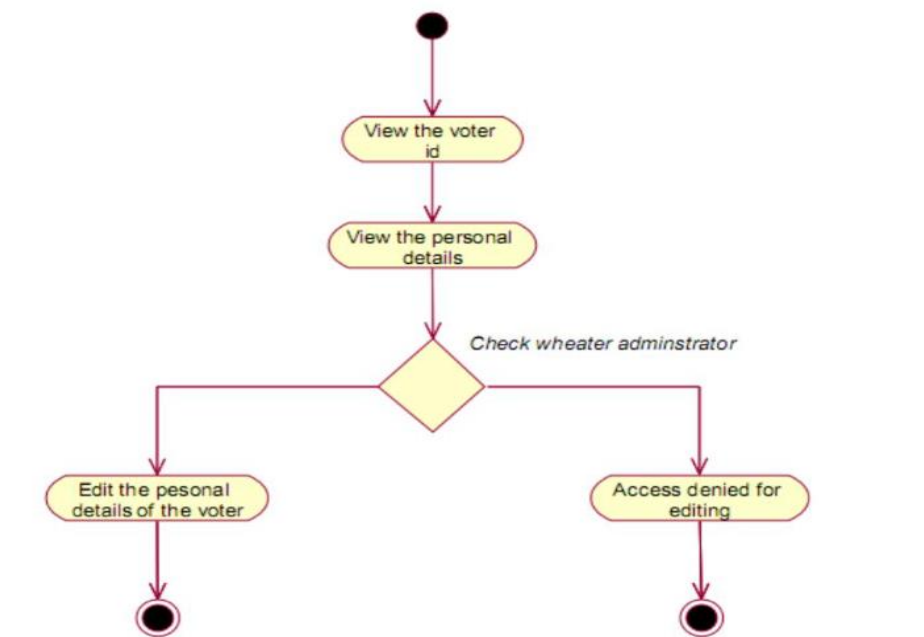
14. Case Diagram



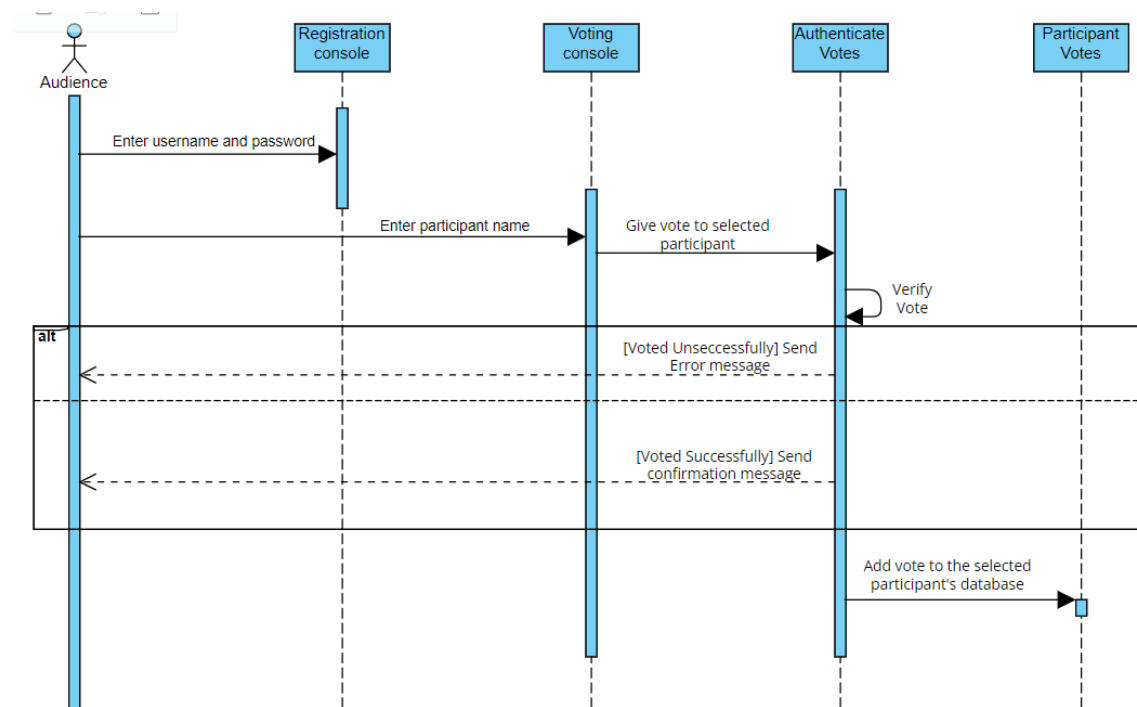
15. Class Diagram



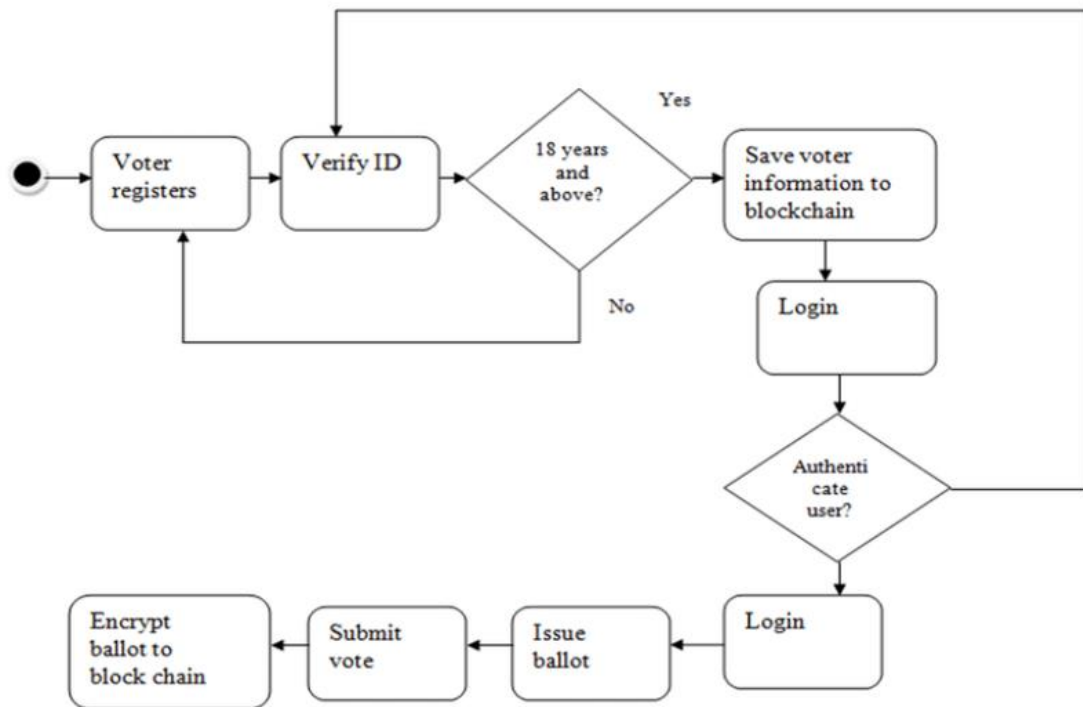
16.Detailed Diagram



17 Sequence Diagram

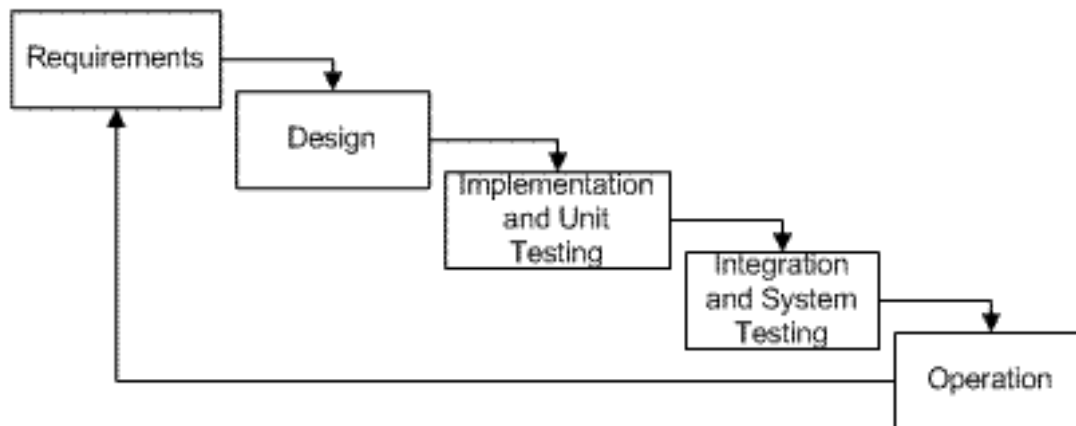


18.State Diagram



19. Different Models

19.1 Incremental Model



Requirements:

- A simple voting system that allows users to vote for a single candidate
- Basic security and privacy measures to protect user information

Design:

- Create a simple user interface that allows users to view the candidates and cast their vote
- Use object-oriented design principles to model the candidates and voting system

Implementation:

- Write the code for the voting system in Java
- Implement basic security measures such as user authentication and data encryption
- Create a user-friendly interface for voters to cast their ballots
- Store data in a secure and reliable database

Unit Testing:

- Write unit tests to verify that individual components of the voting system work as expected
- Test for edge cases and potential security vulnerabilities
- Ensure that all code meets quality standards and best practices

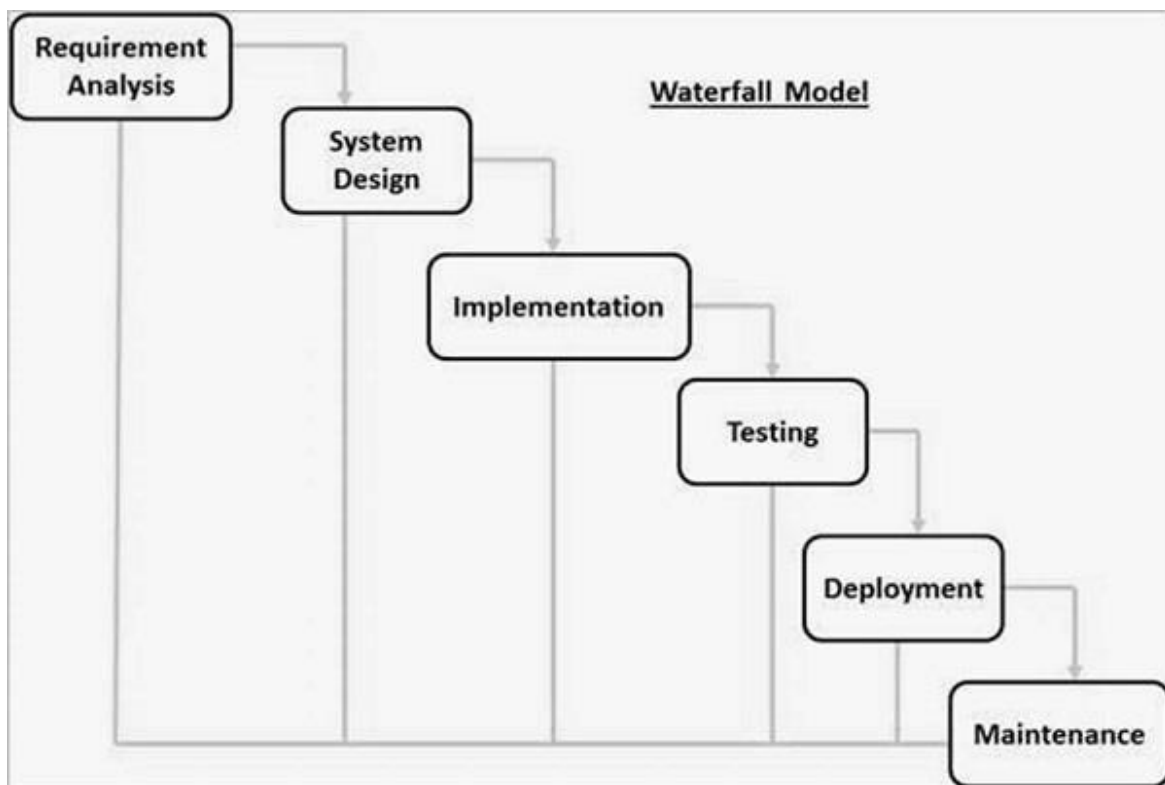
Integration and System Testing:

- Test the integration between the user interface and the database
- Conduct system testing to ensure that the system can handle a limited number of simultaneous users

Operation:

- Deploy the voting system to a secure hosting environment
- Implement monitoring and alerting systems to detect and respond to any issues
- Provide documentation and training for administrators and users
- Regularly review and update the system to address any security vulnerabilities or bugs.

19.2 Waterfall Model



Requirement Analysis:

- Conduct research on the needs and requirements of an online voting system, including user types, features, and security

- Gather feedback from stakeholders, such as voters, election administrators, and government officials
- Define the system's scope, objectives, and functional and non-functional requirements
- Document the findings and produce a detailed requirement specification document

System Design:

- Develop a high-level design of the system that includes the overall architecture, modules, and interfaces
- Create detailed design specifications for each module, including input/output formats, data structures, and algorithms
- Use object-oriented design principles to model the real-world entities such as voters, candidates, and elections
- Ensure that the system design adheres to the requirements and best practices in Java design

Implementation:

- Write the code for the online voting system in Java, following the design specifications and adhering to best practices
- Implement the features and functionalities as specified in the requirement specification document
- Ensure that the system is well-structured, modular, and easy to maintain

Testing:

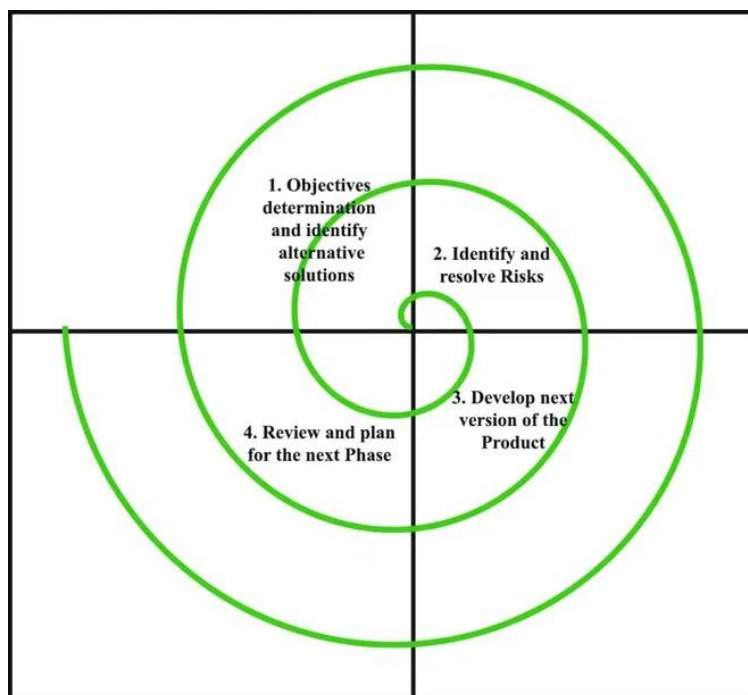
- Test the online voting system for functionality, user interface, and security
- Write unit tests to verify that individual components of the system work as expected
- Test the integration of the modules and the system as a whole
- Ensure that the system meets all functional and non-functional requirements

Deployment:

- Deploy the online voting system to a secure and reliable hosting environment
- Ensure that the system is accessible to the intended users and meets the required availability and performance standards

Maintenance:

- Provide ongoing maintenance and support for the online voting system
- Regularly update the system to address any security vulnerabilities or bugs
- Monitor the system for any performance issues or user feedback and respond accordingly
- Provide training and documentation for administrators and users

19.3 Spiral Model**Objectives Determination and Identify Alternative Solutions:**

- Define the objectives of the online voting system, such as secure authentication, user-friendly interface, and accurate vote counting

- Identify the stakeholders and their needs and expectations for the system
- Explore alternative solutions and technologies, such as blockchain, encryption, and biometric authentication, to ensure the system's security and reliability

Identify and Resolve Risks:

- Identify potential risks and challenges in the development and implementation of the online voting system, such as security breaches, data loss, and system failures
- Develop a risk management plan that includes risk identification, assessment, and mitigation strategies
- Implement preventive measures to reduce the likelihood of risks and ensure business continuity

Review and Plan for the Next Phase:

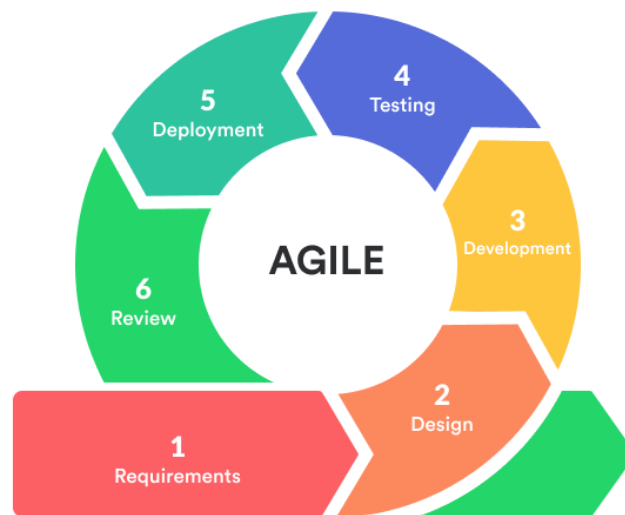
- Review the progress and outcomes of the current phase of the project, including the successes and challenges
- Identify the key objectives and deliverables for the next phase of the project, and develop a detailed plan for implementation
- Allocate resources, assign tasks, and establish deadlines and milestones
- Review the project schedule, budget, and risk management plan, and make any necessary adjustments

Develop Next Version of the Product:

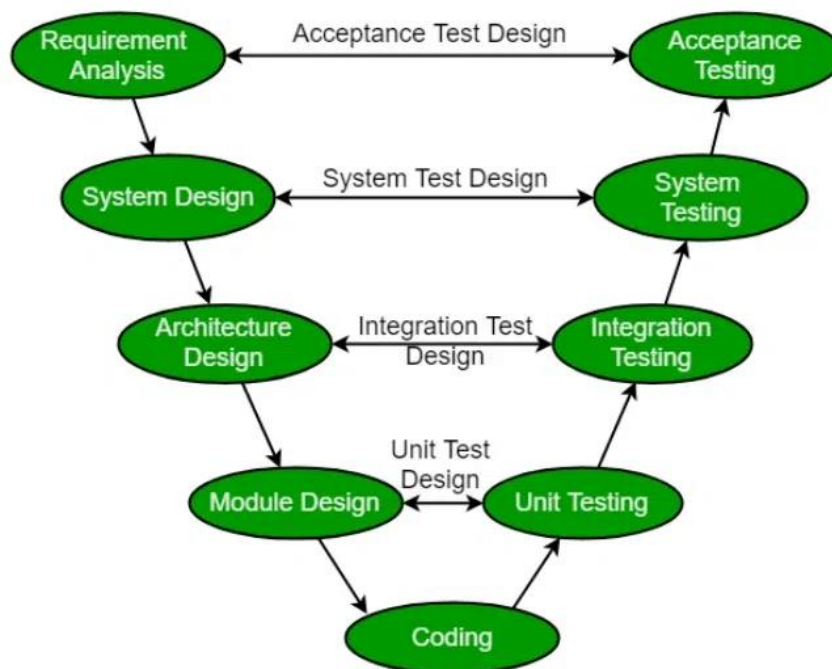
- Develop the next version of the online voting system based on the feedback and requirements from the stakeholders
- Incorporate new features and functionalities, such as voter registration, vote tracking, and result reporting, based on the project objectives and requirements
- Ensure that the system is secure, scalable, and maintainable, and adheres to the best practices in Java design and development
- Test the system thoroughly and address any defects and issues before release

- Provide training and documentation for the users and administrators to ensure a smooth and successful implementation of the online voting system.

19.4 Agile Model



19.5 V-Model



Requirement Analysis: In this stage, we will define the goals and objectives of the online voting system. This includes identifying the features required for the system such as user registration, voting, result viewing, and security measures.

Module Design: Next, we will design the modules required for the system. For example, we can have separate modules for user registration, voting, and result viewing. Each module will have specific design requirements that need to be met.

Unit Test Design: At this stage, we will design the unit tests for each module. This involves identifying the test cases for each module and the expected outcomes.

Unit Testing: In this stage, we will implement the unit tests designed in the previous stage. We will test each module individually to ensure that it functions as expected.

Integration Test Design: After unit testing, we will design the integration tests to ensure that the modules work together seamlessly. This stage involves

identifying the test cases for the system as a whole and the expected outcomes.

Integration Testing: We will implement the integration tests designed in the previous stage. This stage involves testing the system as a whole to ensure that all the modules work together correctly.

System Test Design: In this stage, we will design the system tests to ensure that the system meets the user requirements. We will identify the test cases and the expected outcomes for the system.

System Testing: We will implement the system tests designed in the previous stage. This stage involves testing the system to ensure that it meets the user requirements.

Acceptance Test Design: At this stage, we will design the acceptance tests to ensure that the system is ready for deployment. We will identify the test cases and the expected outcomes for the system.

Acceptance Testing: Finally, we will implement the acceptance tests designed in the previous stage. This stage involves testing the system to ensure that it is ready for deployment and meets the user requirements.

20. Future Scope

The future scope for an online voting system Java mini-project includes enhancing security with encryption and multi-factor authentication, integrating biometric authentication for identity verification, and exploring blockchain technology for transparency. Additional features could include accessibility options, mobile app development, real-time monitoring, and data analysis tools. Integration with government systems for eligibility verification and multi-language support can improve inclusivity. Feedback mechanisms and scalability optimizations ensure continuous improvement and performance under heavy loads. Simulation and testing tools help validate reliability and accuracy, making the system robust for real-world use.

21. Conclusion

In conclusion, the future scope for the online voting system Java mini-project is vast and promising. By prioritizing security enhancements, accessibility features, and integration with emerging technologies like blockchain, the system can evolve to meet the demands of modern elections. Continued development in areas such as mobile applications, real-time monitoring, and data analysis ensures usability and effectiveness. Integration with government systems and feedback mechanisms enhance reliability and inclusivity, while scalability optimizations and rigorous testing guarantee performance and accuracy. With these advancements, the online voting system can become a trusted and indispensable tool for democratic processes worldwide.

22 References

- 1) <https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>
- 2) Software Requirement Specification Documentation by Leslie pinto and Rohit Vanvani
- 3) <https://www.scribd.com/document/367936783/Srs-of-E-Voting-System>
- 4) Software Requirement Specification E-Elections Software “The percies Project”
- 5) Fundamentals of Software Engineering, Fourth Edition, Rajib Mall
- 6) <https://slcc.pressbooks.pub/technicalwritingatslcc/chapter/software-requirements-specification-srs/>