Voting system requirements

NWU

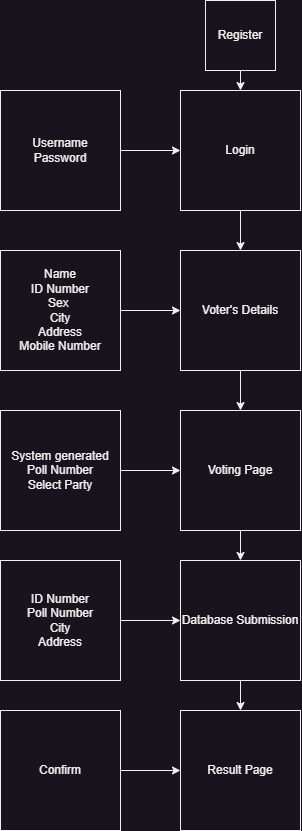
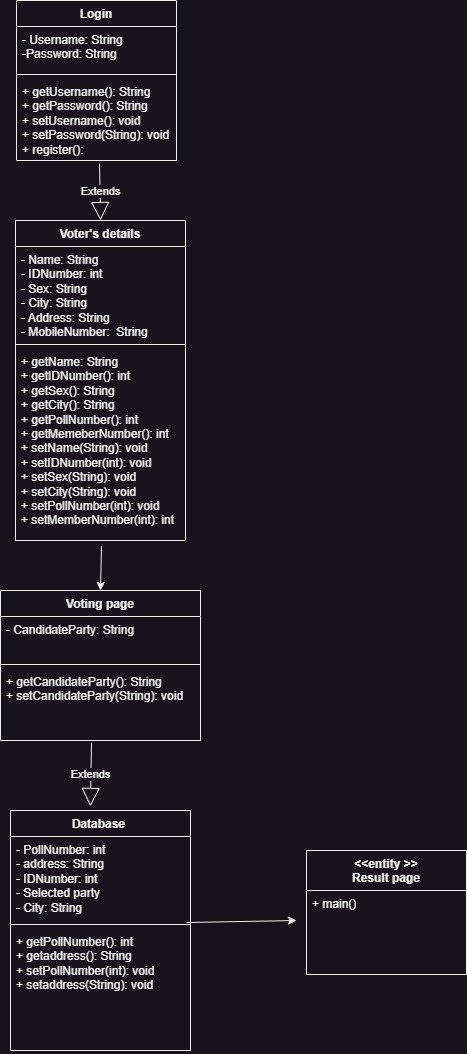
CMPG211

**Table of Contents**

|  |  |
| --- | --- |
| **Content** | **Page Number** |
| **Introduction** | **2** |
| **Planning** | **3-4** |
| **Requirements** | **5-6** |
| **Wireframe of the System** | **7** |
| **Work Breakdown Structure (WBS)** | **8** |
| **Timeline** | **9** |
| **Inheritance, Polymorphism, Encapsulation and Abstraction** | **10-13** |
| **Conclusion** | **14** |

**Voting System Project Documentation**

1. **INTRODUCTION**  
   In this project, we­ are creating a straightforward yet practical voting system for conducting elections. It will allow eligible users to cast their votes electronically, ensuring transparency, accuracy, and ease of use. The system will include features such as voter registration, candidate selection, vote submission, and result tabulation.
2. **PLANNING**

* Block Diagram:  
  ****
* Class Diagram:  
  

1. **Requirements**

* **Functional Requirements:**
* User Registration and Authentication:
* Users must register with valid credentials.
* Authentication mechanisms (e.g., username-password, biometrics) should be implemented.
* Candidate Management:
* Admins can add, edit, and delete candidate profiles.
* Candidates’ information (name, party, photo) should be stored.
* Voting Process:
* Voters can select candidates.
* Votes should be recorded securely and anonymously.
* Prevent multiple voting by the same user.
* Result Calculation:
* Calculate and display real-time voting results.
* Ensure accuracy and integrity of results.
* **Non-Functional Requirements:**
* Security:
* Protect against unauthorized access, data tampering, and fraud.
* Implement encryption for sensitive data.
* Scalability:
* The system should handle a large number of users during peak voting times.
* Scalable architecture and load balancing are essential.
* Develop contingency plans to address potential system failures or disruptions, such as system breakdowns or natural disasters.
* Usability:
* Intuitive user interfaces for voters and administrators.
* Clear instructions for the voting process.
* Ensure compatibility with different operating systems, including Windows, Mac, iOS, and Android, to accommodate a diverse user base.
* Legal and Compliance:
* Ensure compliance with relevant election laws and data privacy regulations.
* Implement measures to protect voter anonymity and privacy.

1. **Wireframe of the System**

A screenshot of a cell phone

Description automatically generated

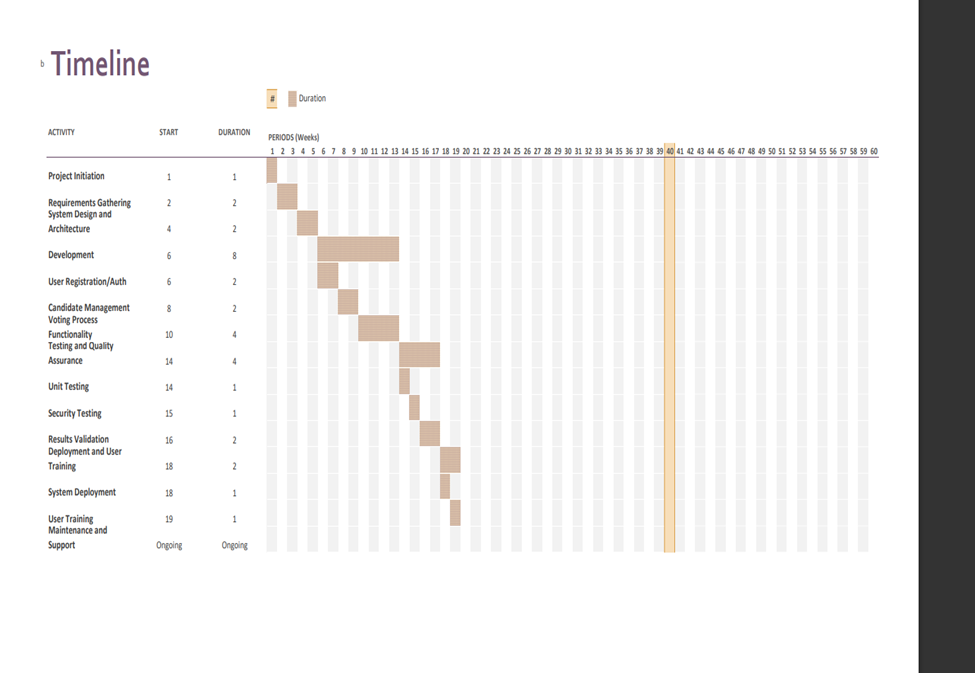
**A screen shot of a cell phone

Description automatically generated**

1. **Work Breakdown Structure (WBS)**

* Project Initiation:
* Define project scope and objectives.
* Form project team.
* Requirements Gathering:
* Collect functional and non-functional requirements.
* Document use cases and user stories.
* System Design and Architecture:
* Create a system block diagram.
* Design database schema.
* Ensure accessibility standards compliance.
* Development:
* Implement user registration and authentication.
* Develop candidate management module.
* Build voting process functionality.
* Testing and Quality Assurance:
* Conduct unit testing.
* Perform security testing.
* Validate results calculation.
* Incorporate User Acceptance Testing.
* Deployment and User Training:
* Deploy the system on servers.
* Train administrators and users.
* Maintenance and Support:
* Monitor system performance.
* Address any issues or enhancements.
* Maintain up-to-date documentation.

1. **TIMELINE**

****

1. **Inheritance, Polymorphism, Encapsulation and Abstraction**

* **Inheritance:**

**A diagram of voting system

Description automatically generated**

**A white background with black text

Description automatically generated**

* The Candidate subclass in the picture above inherit common functionality and attributes from the VotingSystem class, such as the firstName and lastName. This relationship allows for code reuse and promotes a hierarchical structure in the application.
* **Polymorphism:**

**A screenshot of a computer

Description automatically generated**

* Polymorphism allows each subclass to provide its own implementation of the method suited to its specific requirements. In the picture above, the validateUser() method in the Voter class is overridden to implement specific validation logic for voters.
* **Encapsulation:**

A screenshot of a computer

Description automatically generated

* Encapsulation is achieved through the use of access modifiers like private, protected, and public to control the visibility and accessibility of class members. In the picture above, instance variables like firstName and lastName in the VotingSystem abstract class are encapsulated by declaring them as private and providing public getter and setter methods to access and modify them. This encapsulation helps protect the integrity of the data and promotes code maintainability.
* **Abstraction:**

**A screenshot of a computer

Description automatically generated**

* The VotingSystem class in the picture above serves as an abstract representation of a page in the voting system application. This class defines common attributes and methods that are essential for all pages, while leaving the implementation details to its concrete subclasses. By abstracting away the implementation details, the VotingSystemPage class provides a clear and simplified interface for interacting with different types of pages in the application.

1. **Conclusion**

The Voting System project aims to revolutionize the election process by providing a secure, efficient, and user-friendly platform for voters and administrators. By adhering to the requirements and following the project plan, we can achieve a successful implementation while mitigating risks and ensuring compliance with legal and accessibility standards.

**Group members**

**1. Keletso Ramothibe, 45541442 (GL)**

**2. Phomolo Legobye, 41270118**

**3. Total Van Wyk, 45603073**

**4. Nokukhanya Mlilwana, 45756635**

**5. Ndwandwa Ngazibini, 45576483**

**6. Kgalalelo Mokotedi, 38572346**

**7. Nkululeko Mayisa, 43332854**

**8. Olwethu Dingiswayo, 39057399**

**9. Khalishwayo Nkululeko, 40454142**

**10. Thembinkosi Mnchunu, 38208555**