SOFTWARE ENGINEERING

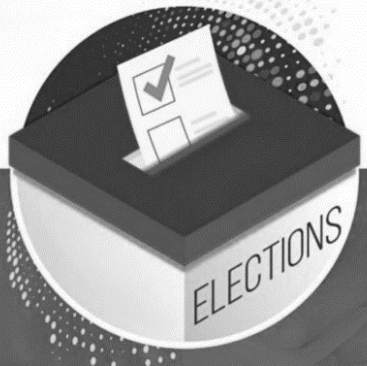
GROUP MEMBERS

SAP-IDs:	Names:
54689	Abdul Rafay
54481	Hassan Zahid
55241	Afaq Ahmed
53143	Ahsan Abbas
Section:	SE 3-1
Submitted to:	Sir Shabbir Hassan
Date:	27th November, 2024

A PROJECT REPORT
ON

ELECTIONS MANAGEMENT SYSTEM (EMS)

FACULTY OF COMPUTING,
RIPHAH INTERNATIONAL UNIVERSITY
I-14 CAMPUS, ISLAMABAD



Project Report Requirements:

- Project Title
- Project brief description.
- Project Scope.
- Functional and Non-Functional Requirements
- Software Design (Class Diagram, Data Flow Diagram whichever is applicable)
- Tools/Technologies.

Project Title:

ELECTIONS MANAGEMENT SYSTEM (EMS)

Project Description:

The Elections Management System (EMS) is a software solution designed to streamline the election process. It provides administrators with the tools to manage voter information, candidate details, and constituencies while ensuring a transparent and fair election. The system includes an Admin Panel for election officials and a Voter Panel for the general public. Voters can access candidate information, stay updated with election news, and securely cast their votes. This system enhances efficiency, security, and user experience in managing elections.

Project Scope:

The EMS aims to facilitate both administrators and voters by offering a centralized platform to handle various election-related tasks. The system will cover the following:

1. **For Administrators**:
   - Voter registration, updates, and removal.
   - Candidate management for NA and PA elections.
   - Constituency organization and management.
   - Real-time election news updates and announcements.
   - Starting and ending election processes.
   - Automated result calculation, display and provide output as a report.
2. **For Voters**:
   - Access to candidate information.
   - Secure platform to cast votes.
   - Real-time updates on election news.

This system ensures transparency, reduces the manual workload, and builds trust in the electoral process.

Functional and Non-Functional Requirements:

- **Functional Requirements:**

1. **Admin Panel Features**:
   - Manage voter information (add, update, delete).
   - Manage candidate details per constituency.
   - Organize constituencies for NA and PA elections.
   - Publish election-related news and updates.
   - Start/End voting process of the election.
   - Automatically calculate and display results.

2. **Voter Panel Features**:
   - Display candidate information for voters' constituencies.
   - Allow secure voting during active elections.
   - Provide election news and updates.

- **Non-Functional Requirements**

1. **Security**:
   - Ensure secure access to the system with role-based authentication.
   - Providing protection to sensitive data like voter information and election results.
2. **Performance**:
   - The system should handle simultaneous access from multiple voters and administrators without performance degradation.
3. **Scalability**:
   - The system should be scalable to accommodate future expansions, such as additional constituencies or voters.
4. **Usability**:
   - Provide a user-friendly interface for both administrators and voters.
   - Ensure the system is easy to navigate and accessible on multiple devices.
5. **Reliability**:
   - The system must be reliable and function without failures during critical periods, such as active voting.
6. **Maintainability**:
   - The system should be easy to update and maintain, with clear documentation for future developers.

Software Design:

The system's design is built on a modular structure, allowing different components to interact seamlessly. The following design elements are central to the EMS:

1. Person (Base Class)

- **Attributes**:
  - name : String — The name of the person (either admin or voter).
  - cnic : String — The unique CNIC number for identification.
- **Description**: This base class serves as a parent class for both Administrator and Voter. It contains attributes common to both.

2. Administrator (Derived from Person)

- **Responsibilities**:
  - Manages voters, candidates, and constituencies.
  - Controls election schedules through the ElectionControl class.
- **Relationships**:
  - Inherits from the Person class.

3. Voter (Derived from Person)

- **Responsibilities**:
  - View candidates and election news.
  - Cast votes during the election.
- **Relationships**:
  - Inherits from the Person class.

4. Candidate

- **Responsibilities**:
  - Stores and manages candidate details added by admin.
  - Tracks votes received by the candidate.

5. Constituency

- **Responsibilities**:
  - Manages the voter count and voter lists for each constituency.

6. ElectionControl

- **Responsibilities**:
  - Manages the schedule and operational status of elections.

7. ResultTabulationOverall

- **Responsibilities**:
  - Calculates and manages overall election results, including turnout and vote statistics in the form of report.

8. ElectionNews

- **Responsibilities**:
  - Displays important updates or notifications to voters and administrators.

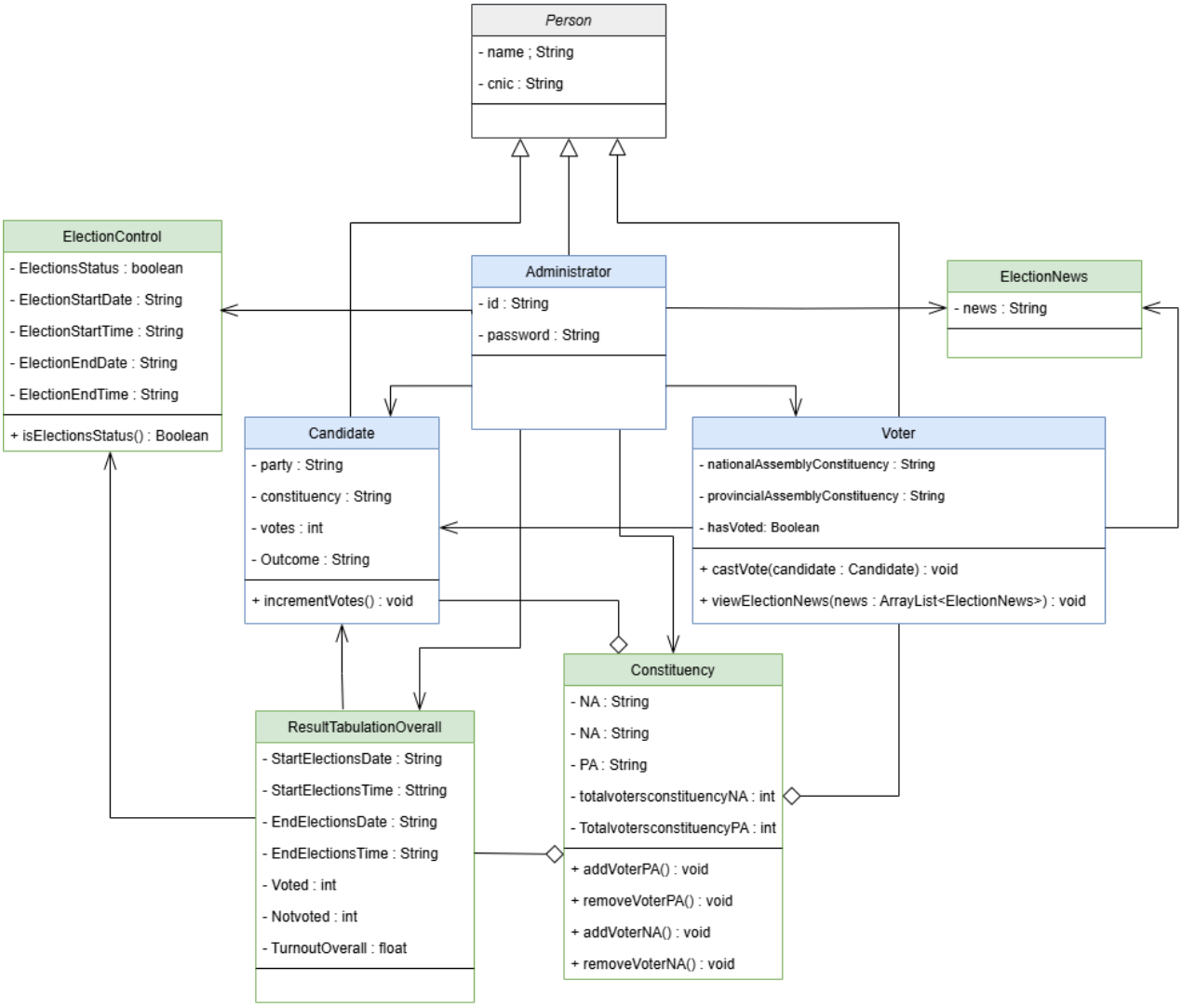


Figure: *Class Diagram of Elections Management System (EMS)*

Relationships in the Diagram

- **Aggregation**:
  - Administrator and Voter interact with ElectionNews to publish and view news, respectively.
  - Administrator and ElectionControl manage election schedules.
- **Composition**:
  - Administrator directly controls Constituency and Candidate management.
  - ResultTabulationOverall depends on data from elections managed by ElectionControl.

Tools/Technologies:

Category	Tools/Technologies
Software Used	NetBeans
Frontend	JSwing (Java)
Backend	Java
Database	No Database (Data stored in ArrayList)

Explanation of Technologies:

- **NetBeans**: An Integrated Development Environment (IDE) that provides all the tools necessary to develop Java applications. It is used for both frontend and backend development of the EMS.
- **JSwing**: A Java library used for building graphical user interfaces (GUIs). It provides components like buttons, panels, and text fields to build the user interface for both the Admin and Voter panels.
- **Java**: A versatile, object-oriented programming language used for implementing the business logic and backend operations of the EMS. It ensures that the system can efficiently process tasks like voting, result calculation, and data management.
- **ArrayList**: Data is stored in memory using ArrayLists, a dynamic data structure in Java, which allows for flexible and efficient handling of data like voters and candidates, without the need for a formal database.

-- ☺ THE END ☺ --