Project Documentation

Submitted To:

Ms. Zoha Sohail

Submitted By:

Zain Khalid 2024-CS-697

Muhammad Ayan Sajid 2024-CS-661

Minahil Ehsan 2024-CS-668

Department of Computer Science

University of Engineering and Technology,

Lahore, New Campus

Contents

Introduction

The Online Voting System is a console-based C++ project designed to allow eligible users to participate in elections electronically. By using object-oriented programming principles like abstraction, inheritance, and polymorphism, it offers a flexible and scalable architecture. The system supports both national and local elections, providing features for administrators, voters, and candidates.

Problem Statement

Traditional paper-based voting systems are time-consuming, error-prone, and inaccessible for many users. This project addresses these issues by offering a secure, digital alternative for conducting elections, ensuring integrity, efficiency, and user-friendly interactions.

Objectives

- Develop a modular, extensible voting system.
- Enable candidate registration and vote tracking.
- Allow voters to securely cast their votes and check status.
- Provide admins tools to manage elections and view results.
- Prevent double voting and ensure transparent vote counting.

System Overview

The system supports three main user roles:

Administrator: Can create/manage elections, add candidates, and view results.

Candidate: Can view elections, set/update their party, and monitor vote counts.

Voter: Can view elections, cast votes, and check voting status.

The application handles multiple election types, each with different rules and workflows. All data persistence is achieved through plain text files.

Key Features

User Authentication

- Secure login using credentials stored in files.
- Only valid users can access functionalities.

Role-Based Access

- Different menus for voters, candidates, and admins.

Election Types

- Supports national and local elections using polymorphism.

Vote Casting

- One vote per user per election.
- Votes are stored and updated in files.

Result Computation

- Vote counts are stored persistently.
- Results are displayed once the election ends.

Class Design

Admin

Attributes: adminProfile

Methods: displayAdminMenu(), userManagement(), partyManagement(), electionManagement(), accountApproval(), viewAdminProfile(), updateCredentialField(), displayUserManagementMenu(), addUser(), deleteUser(), modifyUser(), searchUser(), listAllUsers(), displayPartyManagementMenu(), createParty(), modifyParty(), deleteParty(), listAllParties(), generateUniquePartyID(), isPartyIDUnique(), isPartyNameUnique(), getPartyMemberCount(), getPartyCandidateCount(), displayElectionManagementMenu(), createElection(), modifyElection(), deleteElection(), listAllElections(), manageElectionLifecycle(), viewElectionResultsAdmin(), displayViewAdminProfileMenu(), displayCurrentAdminProfile(), editCurrentAdminProfile(), displayDashboard(), Dashboard(), getProfile(), loadAdminProfile, updateLoginTimestamp()

LoginDashboard (Abstract Class)

Attributes: name, userCNIC

Methods: checkLogin(), login()

AdminLogin (Derived from LoginDashboard)

Methods: checkLogin(), login()

UserLogin (Derived from LoginDashboard)

Attributes: isAlsoCandidate, partyID, isPresident, isDependentCandidate, isParticipating, participatingElectionID

Methods: checkLogin(), login()

SignUpDashboard (Abstract Class)

Methods: signUp(), isValidName(), isValidAge(), isEligible(), isValidCNIC(), isCNICUnique(), isPhoneUnique(), isValidPhoneNumber(), isValidPostalCode(), isValidGender(), isValidGenericText(), isValidPassword(), isValidPartyIDForSignup(), displaySignupHeader(), getBasicInfo(), getDetailedInfo(), getPasswordInfo(), getPartyAffiliation(), signUpHelper(), saveCredentials(), saveData(), addPendingApproval()

UserSignUp (Derived from UserSignUp)

Methods: signUp()

DateTime

Attributes: st, isValid

Methods: padZero(), validateDateTime(), getYear(), getMonth(), getDay(), getHour(), getMinute(), getSecond(), isObjectValid(), toString(), getDateString(), getTimeString(), getCurrentDateTime()

Operators overloaded: =, ==, !=, <, >, <=, >=

Election (Abstract Class)

Attributes: electionID, electionType, statusFromFile, endDateTime, scopeInfo, maxCandidates, winnerCNIC, winnerVoteCount

Methods: setScopeInfo, getElectionID, getElectionType, getStatusFromFile, getEndDateTime, getScopeInfo, getDynamicStatus, getWinnerCNIC, getWinnerVoteCount, setElectionID, setElectionType, setStatusFromFile, setEndDateTime, setWinner, isUserEligibleToVote, isCandidateEligibleToParticipate, getCandidateLimit, getGeographicScope, displayElectionDetails, showCandidatesAndCastVote, concludeAndPublishResults, hasUserVoted, recordUserVote, updateVoteCount, parseDateTimeString, parseElectionLine

NationalElection (Derived from Election)

Methods: isUserEligibleToVote(), isCandidateEligibleToParticipate(), getCandidateLimit(), getGeographicScope(), displayElectionDetails()

RegionalElection (Derived from Election)

Methods: isUserEligibleToVote(), isCandidateEligibleToParticipate(), getCandidateLimit(), getGeographicScope(), displayElectionDetails()

MetropolitanElection (Derived from Election)

Methods: isUserEligibleToVote(), isCandidateEligibleToParticipate(), getCandidateLimit(), getGeographicScope(), displayElectionDetails()

PoliticalParty (Derived from Election)

Attributes: partyID, partyName, rank, membersLimit, candidatesLimit, partyPresidentCNIC

Methods: setLimitsByRank(), setPartyID(), setPartyName(), setPartyRank(), setPartyPresidentCNIC(), getPartyID(), getPartyName(), getPartyRank(), getPartyRankString(), getMembersLimit(), getCandidatesLimit(), getPartyPresidentCNIC(), toString(), fromString(), stringToPartyRank(), partyRankToString()

Profile

Attributes: cnic, firstName, lastName, age, phone, gender, state, city, town, postalCode, lastLoginTimestamp, signupTimestamp

Methods: parseDataLine(), setCnic(), setFirstName(), setLastName(), setAge(), setPhone(), setGender(), setState(), setCity(), setTown(), setPostalCode(), setLastLoginTimestamp(), setSignupTimestamp(), getCnic(), getFirstName(), getLastName(), getFullName(), getAge(), getPhone(), getGender(), getState(), getCity(), getTown(), getPostalCode(), getLastLoginTimestamp(), getSignupTimestamp(), loadFromFile(), updateTimestampInFile()

User

Attributes: userProfile, isRegisteredCandidate_flag, partyID_val, isPresident_flag, isDependentCandidate_flag, isParticipating_flag, participatingElectionID_val

Methods: displayUserMenu(), viewElectionSchedule(), castVote(), viewEditProfile(), viewElectionResults(), electionManagementMenu(), registerInElection(), unregisterFromElection(), partyManagementMenu(), registerInParty(), unregisterFromParty(), viewPartyDetails(), viewPartyMembers(), modifyPartyDetails(), addMemberToParty(), removeMemberFromParty(), setCandidateStatus(), setPartyID(), setIsPresident(), setIsDependentCandidate(), setIsParticipating(), setParticipatingElectionID(), Dashboard(), getProfile(), loadUserProfile(), updateLoginTimestamp()

Object-Oriented Concepts

Abstraction: Abstract base classes like SignUpDashboard and Election expose only relevant functionalities.

Inheritance: Election types inherit from base class Election

Polymorphism: Election handling methods behave differently depending on election type.

Encapsulation: Sensitive data like votes and credentials are accessed only through class methods.

System Flow

- Admin logs in and creates election
- Candidates register and set party
- Voters log in and view active elections
- Voters cast votes

- Election ends, and results are displayed

UML Diagram

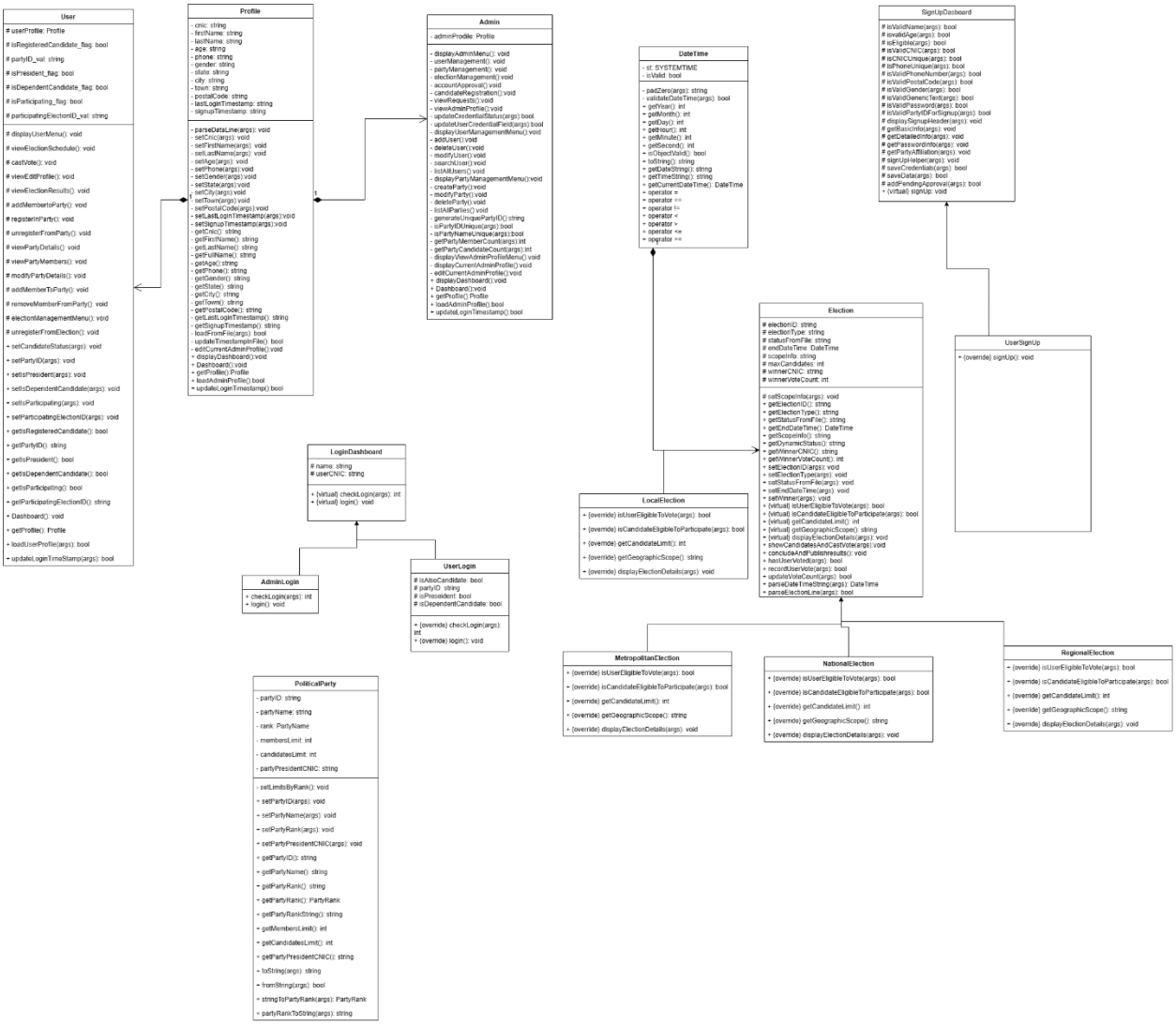

Figure 1: UML Diagram

GitHub Repository

Repository link: https://github.com/MuhammadAyanSajid/Voting-system.git

Conclusion

The "Online Voting System" demonstrates the effective application of OOP in solving real-world problems like digital elections. It supports multiple roles, enforces role-specific access, manages elections, and ensures vote integrity—all via a command-line interface and file persistence.