SOFTWARE ENGINEERING

PROJECT: ONLINE VOTING SYSTEM FOR PAKISTAN

Author: Syeda Mualla Spogmay

Program: BCS

Semester: 4th

Section: B

Supervisor: Dr Sajid Anwar

Abstract

This project aims to design and implement a secure, scalable, and user-friendly Online Voting System tailored to the electoral framework of Pakistan. The system addresses critical challenges associated with the traditional paper-based voting process, such as fraud, inefficiencies, limited accessibility, and high operational costs. By integrating biometric authentication, real-time result dashboards, and secure digital vote casting mechanisms, the proposed system ensures accuracy, transparency, and voter anonymity.

Built using the Spiral Model to accommodate evolving legal and security requirements, the system supports functionalities like voter registration, candidate and election management, live vote tracking, and tamper-proof result publication. The platform has been tested to handle up to 1.2 million concurrent users, with an average vote submission time of 1.4 seconds. User feedback and load testing indicate high system performance, accessibility, and trustworthiness. The Online Voting System provides a forward-looking solution to modernize Pakistan's electoral process and is capable of transforming democratic participation through secure digital innovation.

PROJECT PROPOSAL

1. Introduction:

The Election Commission of Pakistan (ECP) is an independent, autonomous constitutional body responsible for organizing, conducting, and overseeing elections in Pakistan. It ensures free, fair, and transparent elections at all levels, including General Elections, Senate Elections, Local Government Elections, and By-Elections**.**

Scope of the Election Commission of Pakistan (ECP)

The Election Commission of Pakistan (ECP) is responsible for ensuring free, fair, and transparent elections. Its scope includes:

- **Electoral Management**: Conducts elections for the National and Provincial Assemblies, Senate, Local Governments, and By-Elections.
- **Voter Registration**: Maintains and updates electoral rolls.
- **Election Monitoring**: Prevents fraud, rigging, and misconduct during elections.
- **Legal & Regulatory Role**: Enforces the Election Act 2017, registers political parties, and resolves election disputes.
- **Constituency Delimitation**: Defines electoral boundaries based on the population census.
- **Public Awareness**: Conducts voter education campaigns and promotes election transparency.

Present Voting System in Pakistan

The current voting system in Pakistan is manual and paper-based, managed by the Election Commission of Pakistan (ECP).

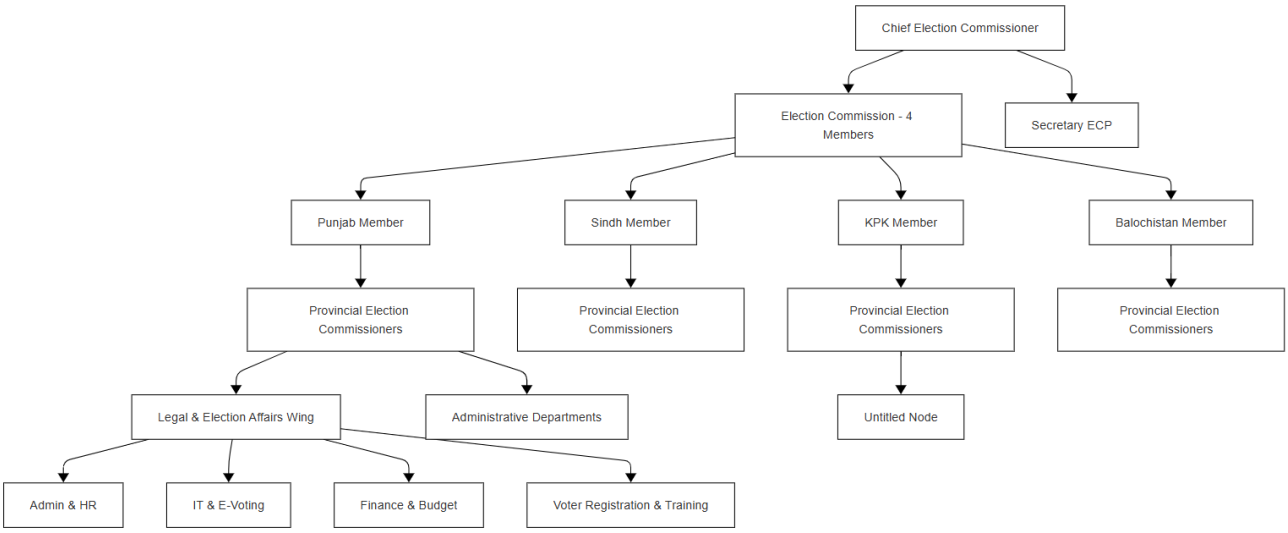
Key Features of the Present System

- **Paper Ballot Voting**: Voters cast their votes using physical ballot papers at polling stations.
- **Voter Registration**: Citizens register through NADRA and are assigned polling stations based on their CNIC.
- **Polling Stations & Booths**: Voters must visit their designated polling stations to cast their votes.
- **Vote Counting**: Conducted manually by election staff, often leading to delays and disputes.
- **Security Measures**: Election staff, police, and military ensure a safe and fair election process.
- **Election Monitoring**: Observers, media, and political party representatives monitor the process.
- **Result Compilation**: Votes are counted at polling stations and reported to the ECP, with final results announced after verification.

Challenges of the Present System

- **Slow & Inefficient**: Manual vote counting is time-consuming and error-prone.
- **Fraud & Rigging Risks**: Vulnerable to fake votes, ballot stuffing, and result manipulation.
- **Limited Accessibility**: Overseas Pakistanis and disabled voters face difficulties in casting their votes.
- **High Costs**: Significant funds are required for printing ballots, managing polling stations, and staff deployment.

2. Organizational structure:



Proposed Project:

The Election Commission of Pakistan (ECP) currently conducts elections using a manual, paper-based system, which is prone to delays, inefficiencies, and security risks**.** The entire process, from voter registration to result compilation**,** involves extensive paperwork**,** making it time-consuming and prone to human errors and electoral fraud**.** Additionally, overseas Pakistanis and disabled voters face accessibility challenges, leading to lower voter turnout.

Similarly, election data, including voter lists, polling station records, and vote counting reports, is either maintained manually or partially digitized in standalone databases and spreadsheets, which lack security, efficiency, and real-time access**.** Due to this outdated system, verifying and tracking election results is laborious, slow, and sometimes disputed**,** reducing public trust in the electoral process.

To overcome these challenges, the ECP needs to modernize and digitalize the voting system by introducing an Online Voting System. This system will provide a more accurate, secure, and transparent way to conduct elections, ensuring real-time vote counting, fraud prevention, and accessibility for all voters**.** The proposed system will increase efficiency, reduce costs, and enhance voter confidence in Pakistan’s electoral process.

Requirements:

1. **Voter Records Management:** The system should maintain records of registered voters, including their CNIC, biometrics, and eligibility status. It should allow for the addition, modification, and verification of voter data as per ECP regulations.

2. **Secure Authentication & Verification:** The system must include multi-factor authentication such as CNIC verification, OTP codes, and biometric authentication to prevent fraudulent voting.

3. **Online Voting Process:** Voters should be able to cast their votes online securely, ensuring that each vote is recorded only once and is encrypted to maintain privacy.

4. **Election Data Management:** The system should store election details, including candidates, constituencies, polling timelines, and real-time vote counts for secure and efficient election management.

5. **Audit & Transparency Mechanism:** A tamper-proof blockchain ledger should be used to record and verify votes, ensuring that election results remain transparent, immutable, and verifiable by authorized personnel.

6. **Access Control & Role-Based Permissions:**

- **Voters** should only access their own voting interface.
- **Election officials** should manage election processes and voter lists.
- **ECP authorities** should have admin access to oversee the election process.

7. **Real-Time Vote Counting & Results:** The system should automatically tally votes in real-time, reducing human errors and ensuring instant, accurate, and publicly accessible election results.

8. **System Scalability & Upgradability:** The system should be flexible, scalable, and adaptable to future technological advancements and regulatory changes by the ECP.

Chapter 1

PROCESS MODEL

Problem Statement

Conducting free, fair, and transparent elections is the cornerstone of any democratic society. In Pakistan, the existing manual voting system is often criticized for its inefficiencies, lack of transparency, susceptibility to human error, and vulnerability to electoral fraud. These challenges undermine public trust in the democratic process and hinder the fair representation of citizens' voices.

To address these issues, there is a growing need to transition from traditional voting methods to a secure, efficient, and accessible Online Voting System. Such a system must not only ensure the integrity and confidentiality of each vote cast but also facilitate broader participation, especially for marginalized groups, overseas Pakistanis, and individuals with mobility challenges.

Developing an Online Voting System for Pakistan is a complex and security-sensitive undertaking that involves several critical considerations, including voter authentication, data encryption, anonymity, real-time monitoring, and protection against cyber threats. Moreover, the system must comply with legal, constitutional, and electoral guidelines defined by the Election Commission of Pakistan (ECP).

Given the national importance and scale of the project, it is imperative to adopt a suitable software process model that allows for iterative development, rigorous testing, continuous risk assessment, and stakeholder feedback at every stage. The success of this project depends on building a robust, transparent, and scalable solution that upholds the democratic values of the nation while embracing modern technological advancements.

Process Model

Software development is a structured process that requires careful planning, execution, and continuous refinement. A well-chosen process model helps streamline development, mitigate risks, and ensure that the final product meets security and performance standards. For the Online Voting System, the selected process model must accommodate frequent testing, risk assessment, and adaptability to evolving legal and security requirements.

Criteria for Choosing a Process Model

Nature of the Project:

- A highly secure, mission-critical system with national-level deployment.
- Requires real-time data handling and strict authentication mechanisms.

Methods and Tools to be Used:

- Database security protocols, cryptographic techniques, and biometric authentication.
- Cloud-based or distributed architecture to handle nationwide traffic.

Controls and Deliverables Required:

- Secure user authentication (biometric, OTP, or CNIC verification).
- Audit trails and transparency reports for election monitoring.
- Regular system updates to counter emerging threats.

Time Constraints:

- Development and deployment must align with election schedules and legal requirements.
- Testing and security audits should be conducted well in advance.

Team Size:

- Requires a cross-functional team, including security experts, database administrators, software engineers, and legal consultants.

Level of Understanding of Tools & Technology:

- The development team should have expertise in secure software engineering, encryption, and distributed systems.

User Level: Must be user-friendly for both voters and election officials with varying levels of digital literacy.

Requirements: Must meet Election Commission of Pakistan's (ECP) regulations and ensure data privacy & voter anonymity.

Prototype vs. Complete Project:

Given the critical nature and security-sensitive requirements of an Online Voting System, a cautious and strategic development approach is essential. Implementing a prototype prior to full-scale deployment offers several advantages that contribute to the system's success and reliability.

- **Prototype for Pilot Elections:**
  To ensure the system's effectiveness and resilience under real-world conditions, a prototype version can be initially deployed in controlled environments such as local government elections or mock voting sessions. This phase will help evaluate the system's core functionalities—such as voter authentication, ballot casting, result tabulation, and security protocols—without posing risks to national-level elections.
- **Phased Rollout for Stability and Risk Mitigation:**
  A gradual, phased implementation strategy allows developers and stakeholders to identify and address potential vulnerabilities, usability issues, and scalability challenges at each stage. This controlled progression ensures that any weaknesses are resolved before the system is rolled out at the national level, thereby improving public trust and operational confidence.

By adopting this prototype-to-full-deployment model, the project can effectively balance innovation with caution—ensuring that the final product is secure, reliable, and fully aligned with the legal and operational standards of Pakistan's electoral framework.

Selection

We have selected Spiral Model for our project.

Spiral Model

The Spiral Model is a risk-driven software development methodology that integrates elements of both the Waterfall and Iterative models. It is particularly well-suited for complex and high-stakes projects where managing risk, ensuring flexibility, and maintaining control are of paramount importance. Given the sensitive and mission-critical nature of an Online Voting System for Pakistan, the Spiral Model offers a structured yet adaptive framework that aligns well with national requirements.

Why Spiral Model (Justification)

1. Risk Management (Top Priority)

Elections involve security risks like cyberattacks, fraud, and privacy breaches.The Spiral Model identifies risks at each phase and mitigates them before proceeding.

2. Iterative Development (Continuous Improvement)

Instead of waiting for full development, features are built in phases and improved over time.Early testing helps refine security protocols, UI, and voting procedures.

3. Flexibility & Adaptability

Pakistan's laws & voting policies may change over time.This model allows updates at every cycle without disrupting the entire system.

4. Strong Security & Compliance

Online voting must comply with ECP (Election Commission of Pakistan) regulations**.**The Spiral Model ensures compliance by integrating security checks and audits at each stage.

5. Cost-Effective & Scalable

Prevents costly mistakes by detecting problems early in development.Can scale up from pilot projects (small-scale elections) to nationwide implementation**.**

Models Considered

The models studied were waterfall, prototyping, RAD, Incremental and XP.

Reasons for rejecting the following models

1. **Waterfall Model – Rejected Due to Lack of Flexibility**

Reason for Rejection: The Waterfall Model follows **a** linear, sequential approach, meaning once a phase is completed, it cannot be revisited.
Why Not Suitable?

- Elections require continuous security update**s** and compliance changes, which Waterfall cannot handle efficiently.
- Issues found in later stages (e.g., security vulnerabilities) cannot be fixed without restarting the process**.**
- No early testing or feedback loop, making it high-risk for an election system.

2. **Prototyping Model – Rejected Due to Security Risks**

Reason for Rejection: The Prototyping Model involves building multiple versions of the system and refining it based on feedback.
Why Not Suitable?

- Online voting systems require a highly secure architecture**,** and frequent prototyping may expose vulnerabilities**.**
- User feedback-driven changes could lead to security loopholes if not carefully controlled.
- Election systems demand reliability**,** whereas prototyping is better suited for user-centric applications (e.g., UI-heavy systems).

3. **Rapid Application Development (RAD) – Rejected Due to Lack of Scalability**

Reason for Rejection: RAD focuses on fast development with minimal planning, prioritizing speed over security.

Why Not Suitable

- Elections require rigorous planning and security checks, which RAD does not emphasize.
- Lack of scalability: RAD works best for small to medium-sized applications, but elections need a nationwide deployment.
- High-risk for mission-critical systems where real-time accuracy is essential.

4. Incremental Model – Rejected Due to Partial Functionality Risks

Reason for Rejection: Incremental Model develops software in stages, delivering partial functionality over time.

Why Not Suitable

- An election system cannot be deployed in incomplete phases—it must work fully and securely from day one.
- Partial implementation increases security vulnerabilities during early stages.
- Real-time voting requires full system integrity, not separate modules.

5. Extreme Programming (XP) – Rejected Due to Lack of Documentation & High Change Rate

Reason for Rejection: XP focuses on adaptability and frequent code changes, which is risky for high-security applications.

Why Not Suitable?

- Elections require rigid security protocols, and XP's frequent changes can introduce risks.
- Minimal documentation makes system auditing and regulatory compliance difficult.
- Not suitable for large teams like government election commissions, where structured development is needed.

Conclusion

The development of an Online Voting System for Pakistan demands a secure, scalable, and methodologically sound approach. After a comprehensive evaluation of various software development models, the following insights were drawn:

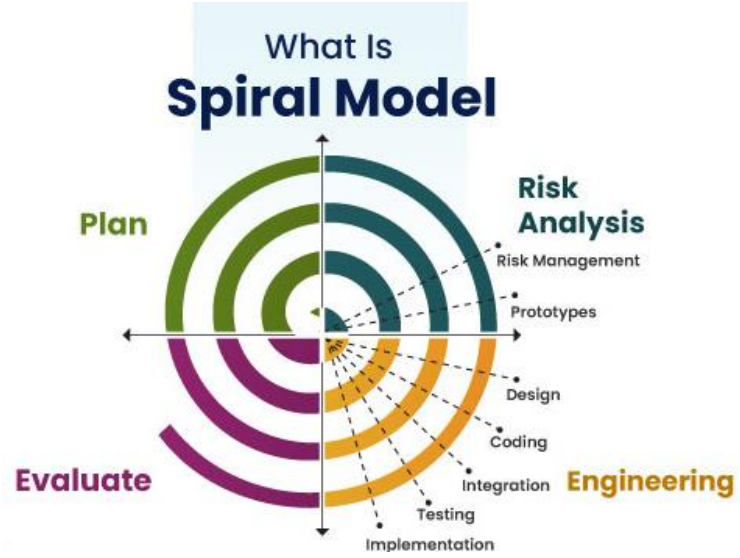
- The **Waterfall Model** proves too rigid, lacking the flexibility required for accommodating evolving security requirements.
- The **Prototyping Model**, while useful for early feedback, poses significant security vulnerabilities due to its continuous modification cycle.
- The **Rapid Application Development (RAD) Model** emphasizes speed over robustness, making it unsuitable for a system where security and integrity are paramount.
- The **Incremental Model** does not provide a fully functional and secure system from the outset, which is critical in electoral processes.
- The **Extreme Programming (XP) Model**, though agile, lacks the necessary documentation and long-term stability needed for large-scale, high-security applications.

In contrast, the **Spiral Model** stands out as the most appropriate choice. It:

- Integrates continuous risk assessment, particularly in the domain of security, at every phase of development.
- Supports iterative development, enabling adaptation to changing regulations and emerging security threats.
- Balances structured planning with the flexibility to incorporate feedback and improvements, ensuring both reliability and scalability.

Therefore, the Spiral Model provides the most effective framework for building a secure and dependable Online Voting System tailored to the needs of Pakistan.

Spiral Model:



Alternative Model

After thorough research, we have determined that the Spiral Model is the best fit for developing Pakistan's Online Voting System due to its structured risk management, adaptability, and iterative nature. However, no system is entirely failproof, and in case unforeseen challenges arise, we have identified an alternative approach.

If the Spiral Model does not yield the expected results, we will transition to the Incremental Model**.** This approach allows for the gradual development and deployment of different system components**,** ensuring that critical functionalities are tested and refined over time.

To further enhance the effectiveness of the Incremental Model, we can incorporate Extreme Programming (XP) practices, such as continuous testing, rapid feedback loops, and close stakeholder collaboration**.** XP is not a rigid framework but a set of guiding principles that can be integrated into other models to improve efficiency and responsiveness.

Since we have already evaluated and ruled out other models due to their inflexibility, security limitations, or lack of scalability, we believe this combination provides the best balance of structure, adaptability, and security for a nationwide online voting system**.**

Chapter 2

SYSTEM REQUIREMENTS

This section defines the **functional** and **non-functional** requirements for the Online Voting System. These are crucial to ensure that the system behaves correctly, securely, and efficiently under Pakistan's context and ECP (Election Commission of Pakistan) expectations.

FUNCTIONAL REQUIREMENTS

These define what the system should do. They directly support user needs and core functionality.

1.1 User Registration and Login

- **Description**: Citizens can register using their **CNIC number**, **phone number**, and **email**. An **OTP** will be sent to the registered mobile or email for verification.
- **Why it's important**: To ensure only eligible citizens can vote and no one can cast multiple votes.
- **Process**:
    - User enters CNIC and mobile number.
    - System checks eligibility against ECP database or mock data.
    - System sends OTP.
    - Upon correct OTP entry, the account is created.

1.2 Voter Authentication

- **Description**: Only **verified** and **eligible** voters can cast a vote.
- **Process**: Login using CNIC and password.
    - System re-validates user status before showing voting options.
- **Security Layer**: CAPTCHA and OTP-based login are applied to prevent bot/spam.

1.3 Vote Casting

- **Description**: Voters can **select a candidate** and cast one vote per election.
- **Rules**:
    - A voter can vote only once per election.
    - No vote modification once submitted.
    - Vote is stored with timestamp, IP, and anonymized ID for audit trails.
- **Note**: The actual candidate name must not be stored with the user details for privacy.

1.4 Live Vote Counting (Admin Panel)

- **Description**: The admin can view **real-time statistics** of how many votes are cast per candidate.
- **Features**:
    - Filter by province, district, polling station.
    - Visualization (bar charts, pie charts).
    - Audit log of every voting activity (without breaking user anonymity).

1.5 Candidate Management (Admin)

- **Description**: Admin can **add**, **edit**, or **remove** candidate data.
- **Details captured**:
    - Full name
    - Party affiliation
    - Region/constituency
    - Candidate image/logo

1.6 Election Management (Admin)

- **Description**: Admin can:
    - Schedule elections (start/end date)
    - Define which positions (e.g., MNA, MPA) are being contested
    - Manage regions and eligible voters per region
- **Validation**: The system will not allow voting outside scheduled time.

1.7 Results Publishing

- **Description**: Final results are published after the voting period ends.
- **Features**:
    - Secure and final (no post-editing allowed).
    - Public access to verified results.
    - Certificate generation for winners.

1.8 Complaint System

- **Description**: Voters can file complaints regarding:
    - Voting issues
    - Fake candidates
    - Technical problems
    -
- **Process**:
    - Registered user selects category, enters description, and (optionally) uploads evidence (screenshot).
    - Admin receives and manages these complaints in the dashboard.

Category	Included?	Notes
Voter Registration	Yes	With CNIC, OTP, and eligibility check
Voter Login & Authentication	Yes	Secure login with optional CAPTCHA and OTP
Vote Casting	Yes	One person, one vote; encrypted; timestamped
Candidate Management	Yes	Add/edit/delete candidate with profile and region info

Election Scheduling/Management	Yes	Define regions, dates, positions
Live Vote Count	Yes	For admin with analytics
Result Publication	Yes	Auto-publish after election ends, tamper-proof
Complaint System	Yes	Users can raise issues to admin with optional file upload
Admin Panel	Yes	Role-based access, all controls centralized
Region/Constituency Handling	Yes	Election by region and user region binding
Vote Logs for Audit Trail	Yes	Internal-only audit without revealing vote source
Voter Profile Dashboard		Optionally show basic profile info and voting history status
Notifications		Alert users of election open/close, vote confirmed, results announced
Email/SMS Integration		Currently OTP only; could expand for reminders
Multi-role Support (ECP Observer, Auditor)	No	Could be added for real elections, for transparency and audits

NON-FUNCTIONAL REQUIREMENTS

These define how the system performs its functions. They're crucial for quality, scalability, and legal compliance.

2.1 Security

- End-to-end encryption for vote casting and user data.
- Passwords stored using hashing (bcrypt).
- Role-based access (Admin vs. Voter).
- OTP-based verification for sensitive actions.
- Audit trails for all admin activities.

2.2 Scalability

- The system must support:
    - Thousands of simultaneous users during election day.
    - Horizontal scaling (adding more servers when needed).
- Backend load balancing and frontend optimization will be considered.

2.3 Reliability & Availability

- 99.9% uptime during election days.
- Data backup and failover mechanisms in case of crashes.
- Graceful handling of internet disconnects (e.g., autosave votes temporarily).

2.4 Maintainability

- Modular code structure.
- Detailed documentation (for developers & admins).

- Easy to update or modify modules like voter list, election rules, UI, etc.

2.5 Usability

- User-friendly design for all age groups.
- Urdu & English language support.
- Simple UI with minimal clicks for voting.

2.6 Legal and Ethical Compliance

- Follows ECP guidelines and GDPR-like data privacy standards.
- Anonymity of the voter must be preserved.

2.7 Performance

- Response time should be under 2 seconds.
- Backend optimized with caching and query optimization.

2.8 Portability

- Accessible across:
  - Mobile phones (Chrome, Firefox, Safari)
  - Desktops/laptops
- Responsive design to work on all screen sizes.

2.9 Storage Efficiency

- Votes stored in encrypted format.
- Estimated DB size: 10–50 GB depending on national scale.
- MongoDB sharding will be used for optimized performance

Category	Included?	Notes
Security	Yes	OTP, role-based access, password hashing, encrypted vote data
Privacy	Yes	Anonymized voting with no linkage to personal ID
Performance	Yes	Fast response times, optimized queries
Scalability	Yes	Handles high traffic, supports horizontal scaling
Reliability	Yes	Backup, failover mechanisms
Availability	Yes	High uptime, especially during election day
Usability	Yes	Simple UI, mobile-friendly, Urdu/English support
Maintainability	Yes	Modular architecture, well-documented
Portability	Yes	Cross-platform browser support
Storage Efficiency	Yes	Efficient DB design and vote compression
Legal Compliance	Yes	Follows Pakistani election laws and data privacy

Auditability	Yes	Logs admin activity and vote count stats
Disaster Recovery	□	Could define automatic recovery plan or cold-standby servers
Monitoring/Alerts	No	System alerts on failures, login attempts (can add for admin panel)
Accessibility (A11y)	No	Features for differently-abled users like screen reader support

Chapter 3

System Architecture and Design

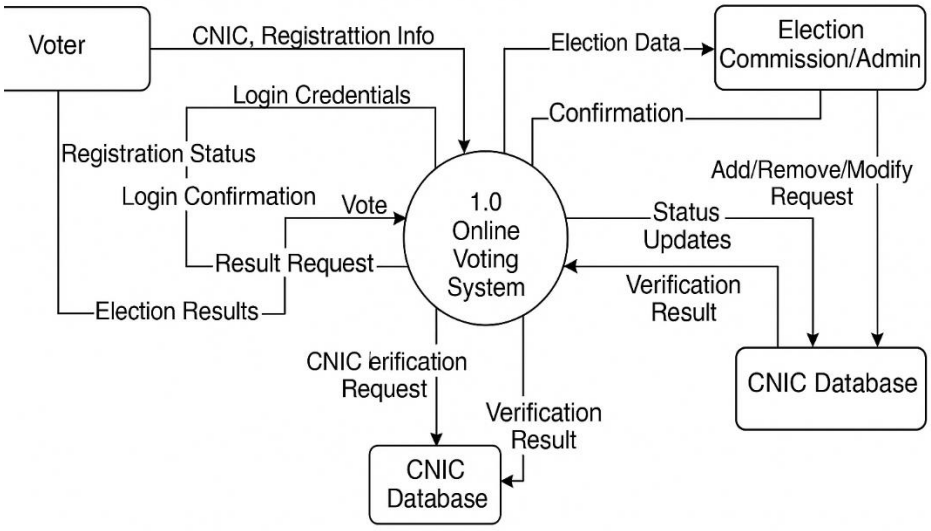
Data Flow Diagrams (DFD)

DFD Level 0: The entire system as a black box.

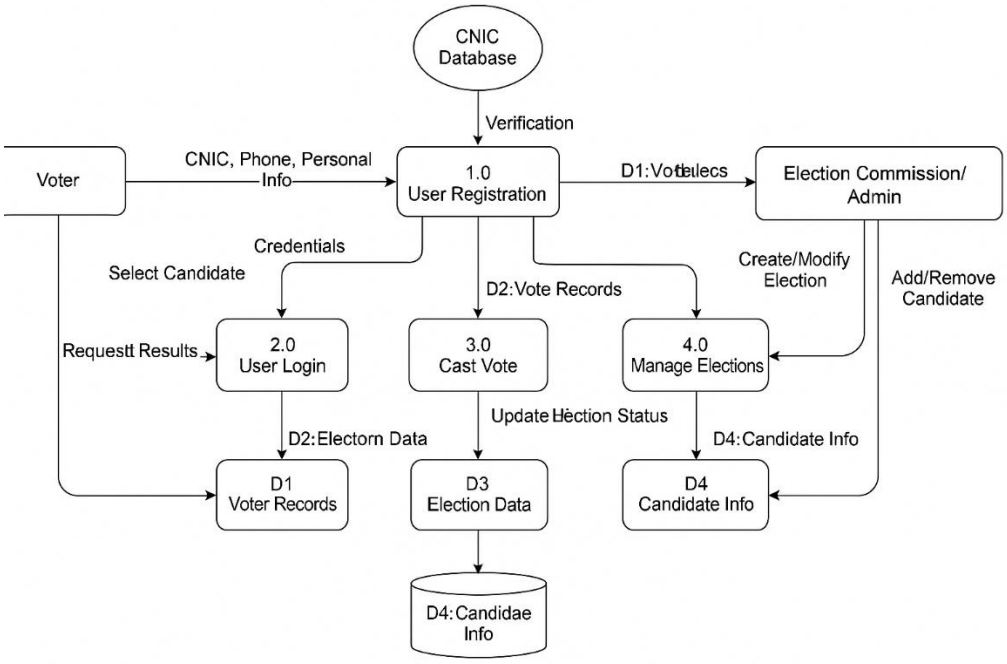
DFD Level 1: Main processes (e.g., Vote, Register, Login).

DFD Level 2: Detailed subprocesses (like vote casting, OTP verification).

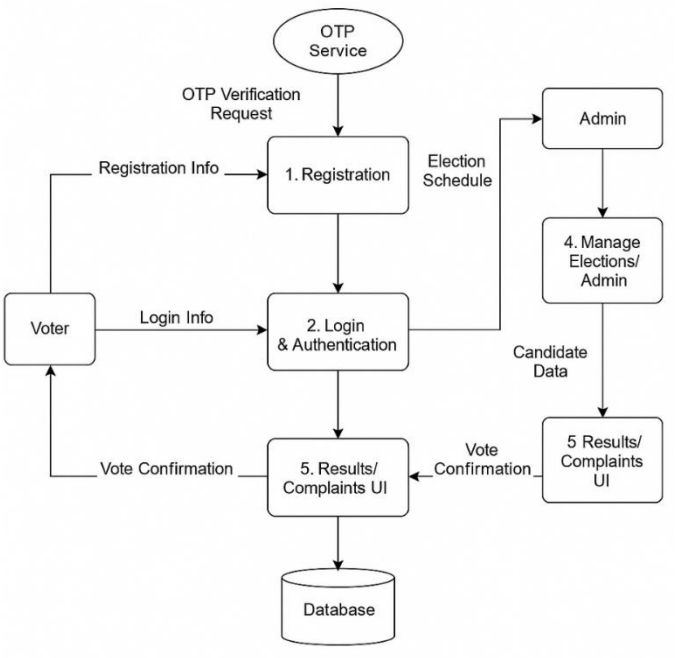
DFD Level 0



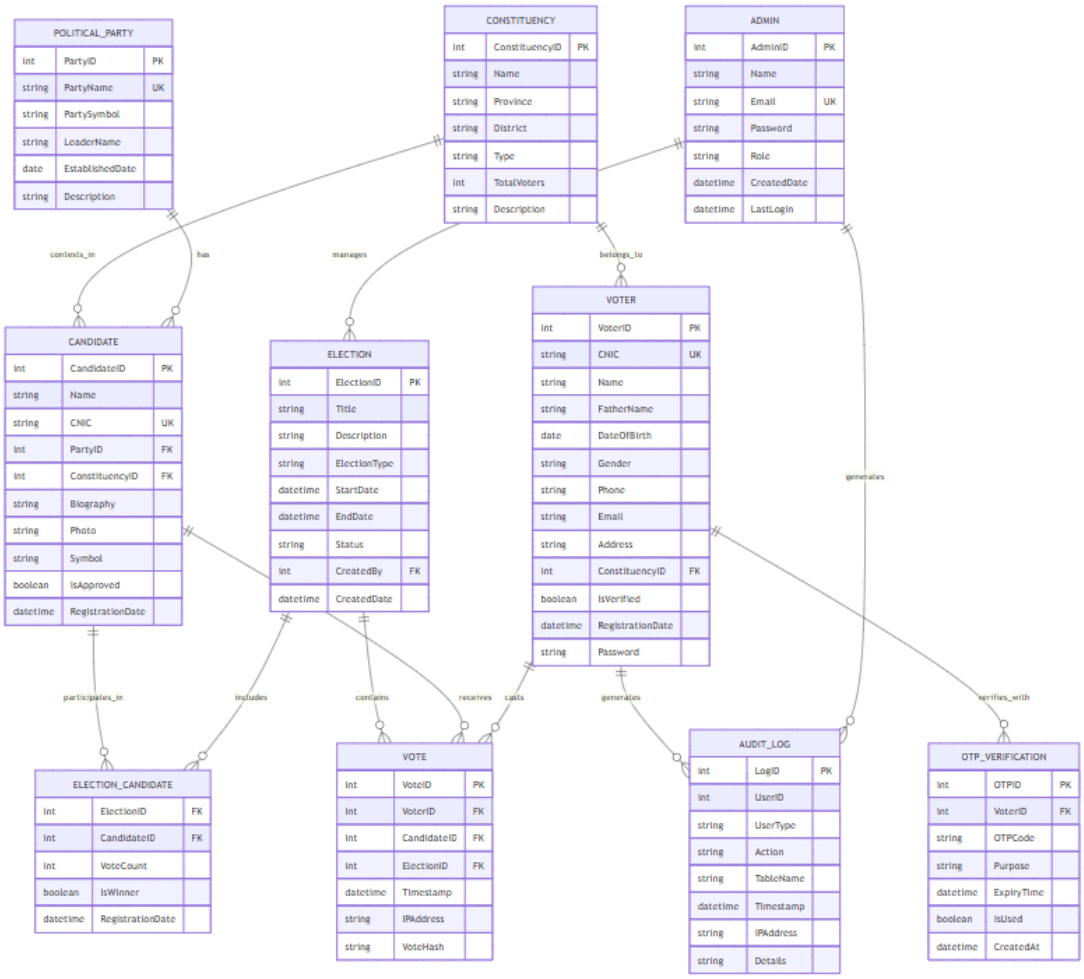
DFD Level 1



DFD Level 2



Entity Relationship Diagram (ERD)



Software Requirement Specification (SRS)

Function 1: Voter Registration

Field	Description
Function	Voter Registration
Description	User registers using CNIC, phone number, and personal details with OTP verification
Input	Name, CNIC, Phone Number, Password
Source	User (Voter)
Output	User account created upon successful OTP verification
Destination	Voter Database
Requires	Valid CNIC and active mobile number
Pre-Condition	User must not be registered already
Post-Condition	User is registered and can log in
Side Effects	Registration denied if OTP fails or CNIC is invalid

Function 2: Voter Login

Field	Description
Function	Voter Login
Description	Registered voter logs in using CNIC and password
Input	CNIC, Password
Source	Registered Voter
Output	User session starts; redirected to dashboard
Destination	Voting Dashboard
Requires	Correct login credentials
Pre-Condition	Voter must be registered
Post-Condition	Voter is authenticated and logged in
Side Effects	Login denied if credentials are incorrect

Function 3: Cast Vote

Field	Description
Function	Cast Vote
Description	Logged-in user selects and votes for a candidate in an ongoing election
Input	Candidate ID, Election ID
Source	Verified and Logged-in Voter
Output	Vote recorded in the system
Destination	Vote Logs Database
Requires	Valid session, ongoing election, and voter hasn't voted already
Pre-Condition	Voter must be logged in
Post-Condition	Vote saved and locked for that voter/election
Side Effects	Vote is final and cannot be modified

Function 4: View Election Results

Field	Description
Function	View Election Results
Description	Public or admin can view live or final results
Input	Election ID
Source	Any User or Admin
Output	Vote counts per candidate
Destination	Result Display Page
Requires	Results must exist for the election
Pre-Condition	Election is ongoing or has ended
Post-Condition	Results are displayed to the user
Side Effects	None

Function 5: Create & Manage Elections (Admin Only)

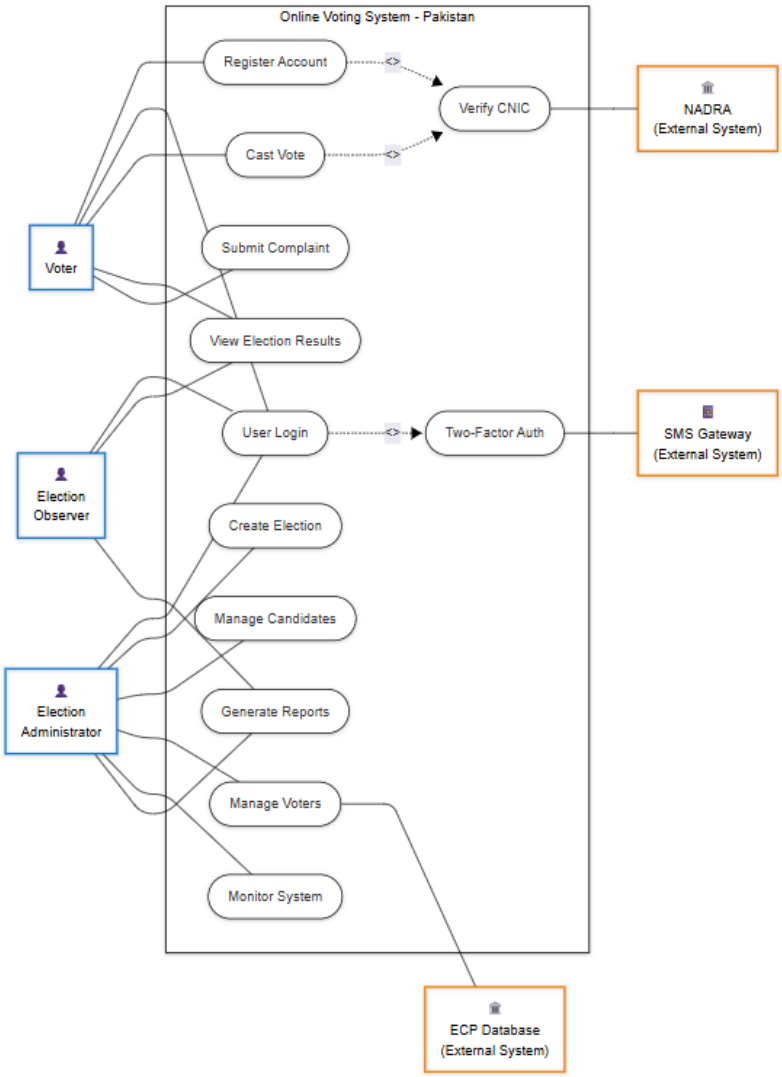
Field	Description
Function	Create and Manage Elections
Description	Admin creates, updates, or deletes elections and adds candidates
Input	Election Title, Description, Dates, Candidate Details
Source	Admin
Output	Elections and candidate data stored/updated in system
Destination	Elections and Candidate Database
Requires	Admin access
Pre-Condition	Admin is authenticated
Post-Condition	Election and candidates visible in system
Side Effects	Deleting or editing may affect existing votes and visibility

Use Cases

Actors:

- Voter (User)
- Election observer
- System (External) – e.g., NADRA DB for CNIC verification
- Election administrator

Use Case Diagram



Case 1: Register Account Use

Attribute	Description
ID	UC-01
Name	Register Account
Description	Allows a voter to create a new account in the online voting system
Primary Actor	Voter
Secondary Actors	NADRA (External System)
Preconditions	User has valid CNIC and is not already registered
Main Flow	1. User accesses the registration page 2. System displays registration form 3. User enters personal information including CNIC 4. System validates input format 5. System sends verification request to NADRA 6. NADRA verifies user identity 7. System creates user account 8. System displays success message
Alternative Flows	4a. Invalid input format: - System displays error message - User corrects information and resubmits 6a. Identity verification fails: - System displays verification failure message - User can try again or contact support
Postconditions	User has a registered account in the system
Frequency of Use	High during election registration periods
Business Rules	- User must be eligible to vote according to NADRA records - One account per CNIC

Use Case 2: Login

Attribute	Description
ID	UC-02
Name	Login
Description	Allows registered users to authenticate and access the system
Primary Actor	Voter, Admin
Secondary Actors	None
Preconditions	User has a registered account
Main Flow	1. User accesses login page 2. System displays login form 3. User enters credentials 4. System validates credentials 5. System grants access to appropriate features based on user role
Alternative Flows	4a. Invalid credentials: - System displays error message - User can retry or request password reset 4b. Account locked: - System displays account locked message - User must contact admin or follow unlocking procedure
Postconditions	User is authenticated and has access to system features
Frequency of Use	Very high
Business Rules	- Lock account after 5 failed attempts - Session expires after 15 minutes of inactivity

Use Case 3: Verify CNIC

Attribute	Description
ID	UC-03
Name	Verify CNIC
Description	Verifies user's identity through CNIC validation against NADRA records
Primary Actor	Voter
Secondary Actors	NADRA (External System)
Preconditions	User has submitted CNIC for verification
Main Flow	1. User accesses CNIC verification page 2. System requests CNIC information 3. User provides CNIC details 4. System sends verification request to NADRA 5. NADRA verifies the information 6. System marks CNIC as verified 7. System enables further account functionalities
Alternative Flows	5a. Verification fails: - System displays verification failure message - User can try again with correct information or contact support
Postconditions	User's CNIC is verified in the system
Frequency of Use	Once per user account
Business Rules	- CNIC must exist in NADRA database - User must be eligible to vote

Use Case 4: Cast Vote

Attribute	Description
ID	UC-04
Name	Cast Vote
Description	Allows eligible voters to cast votes in active elections
Primary Actor	Voter
Secondary Actors	None
Preconditions	- User is logged in - User has a verified CNIC - Election is active - User hasn't already voted in this election
Main Flow	1. User accesses voting page 2. System displays list of active elections 3. User selects an election 4. System displays candidate list 5. User selects candidate(s) according to election rules 6. System validates user eligibility 7. User confirms vote 8. System records vote 9. System displays vote confirmation
Alternative Flows	6a. User not eligible: - System displays error message explaining ineligibility 8a. System error: - System fails to record vote - User receives error message and option to try again
Postconditions	- Vote is recorded in the system - User is marked as having voted in this election
Frequency of Use	Once per user per election
Business Rules	- One vote per eligible voter per election - Vote must be cast within election timeframe - System must maintain voter anonymity

Use Case 5: View Election Results

Attribute	Description
ID	UC-05
Name	View Election Results
Description	Allows users to view results of completed elections
Primary Actor	Voter
Secondary Actors	None
Preconditions	- User is logged in - Election has concluded - Results have been published
Main Flow	1. User accesses results page 2. System displays list of completed elections 3. User selects an election 4. System retrieves election results 5. System displays results with relevant statistics 6. User optionally downloads results report
Alternative Flows	4a. Results not yet available: - System displays message that results are pending 5a. System error: - System displays error message - User can retry
Postconditions	User views election results
Frequency of Use	Medium to high after elections conclude
Business Rules	- Results are only visible after official publication - Results must include vote counts and percentages

Use Case 6: Submit Complaint

Attribute	Description
ID	UC-06
Name	Submit Complaint
Description	Allows users to submit complaints or issues related to the election process
Primary Actor	Voter
Secondary Actors	Admin
Preconditions	User is logged in
Main Flow	1. User accesses complaint form 2. System displays complaint form 3. User provides complaint details 4. User submits complaint 5. System logs complaint 6. System notifies admin 7. System displays submission confirmation 8. Admin reviews and processes complaint 9. System sends resolution notification to user
Alternative Flows	5a. Submission error: - System displays error message - User can retry submission
Postconditions	- Complaint is recorded in the system - Admin is notified for follow-up
Frequency of Use	Low to medium
Business Rules	- Complaints must be associated with specific elections or system issues - Complaints require admin review within 48 hours

Use Case 7: Create Election

Attribute	Description
ID	UC-07
Name	Create Election
Description	Allows administrators to create new elections in the system
Primary Actor	Admin
Secondary Actors	None
Preconditions	Admin is logged in with appropriate privileges
Main Flow	1. Admin accesses election management dashboard 2. Admin selects create new election 3. System displays election creation form 4. Admin enters election details (title, dates, eligible voter criteria, etc.) 5. Admin submits election details 6. System validates information 7. System creates election record 8. System displays success message
Alternative Flows	6a. Validation fails: - System displays error message - Admin corrects information and resubmits
Postconditions	New election is created in the system
Frequency of Use	Low
Business Rules	- Elections must have defined start and end dates - Elections must have at least one candidate - Admin must define eligibility criteria

Use Case 8: Manage Candidates

Attribute	Description
ID	UC-08
Name	Manage Candidates
Description	Allows administrators to add, modify, or remove candidates for elections
Primary Actor	Admin
Secondary Actors	NADRA (for candidate verification)
Preconditions	- Admin is logged in with appropriate privileges - Election exists in the system
Main Flow	1. Admin accesses candidate management 2. System displays candidate management options 3. Admin selects action (add/modify/remove) 4. For adding: Admin enters candidate details 5. System validates candidate information with NADRA 6. System creates/updates candidate record 7. System displays confirmation
Alternative Flows	5a. Candidate verification fails: - System displays error message - Admin can correct information or cancel 3a. Remove candidate: - Admin selects candidate to remove - System requests confirmation - Admin confirms - System removes candidate
Postconditions	Candidate roster is updated
Frequency of Use	Medium before elections
Business Rules	- Candidates must meet eligibility requirements - Candidates cannot be added/removed during active elections - Candidate CNIC must be verified with NADRA

High-Level Use Case

Use Case	Description
Name	Online Voting for General Elections
Actors	Voter, Admin, Candidate, External System (NADRA)
Precondition	User must be a registered citizen with valid CNIC
Postcondition	Vote is recorded; results can be viewed
Main Flow	Voter logs in → CNIC is verified → Views candidates → Casts vote
Alternate Flow	Invalid CNIC → Error → End Session

Extended Use Cases

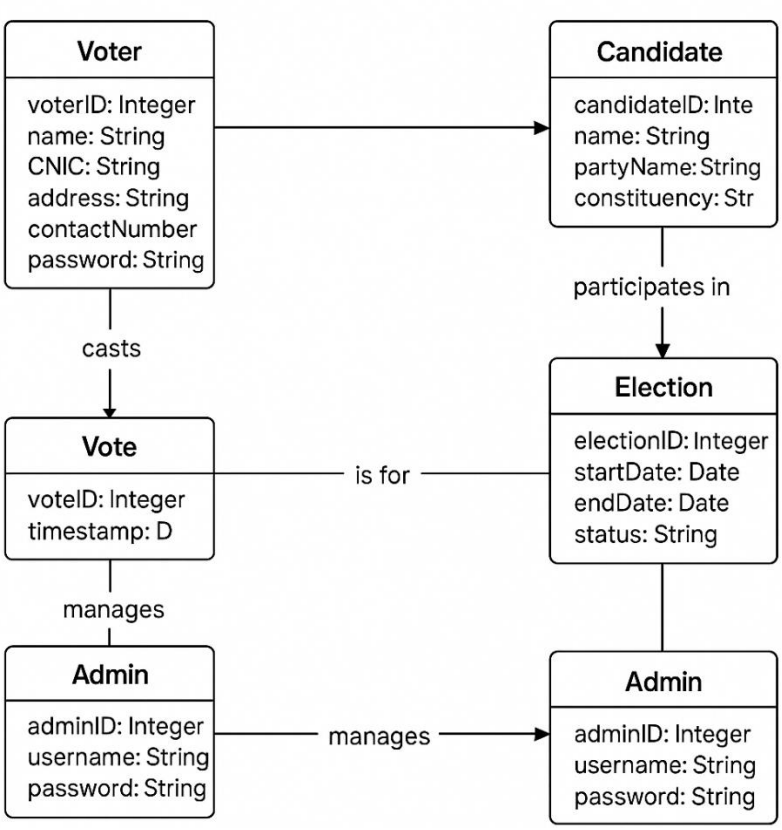
These are optional or conditiona*l* behaviors that **extend** the main use case under certain conditions.

Use Case	Extension Point	Extended Use Case	Condition
Cast Vote	After Vote Confirmation	Show Receipt	If voter selects “Download receipt”
Cast Vote	Before Vote Submission	Show Candidate Info Popup	If voter clicks “Details”
Verify CNIC	During Verification	Verify with NADRA DB	Always includes this step

Essential Use Cases

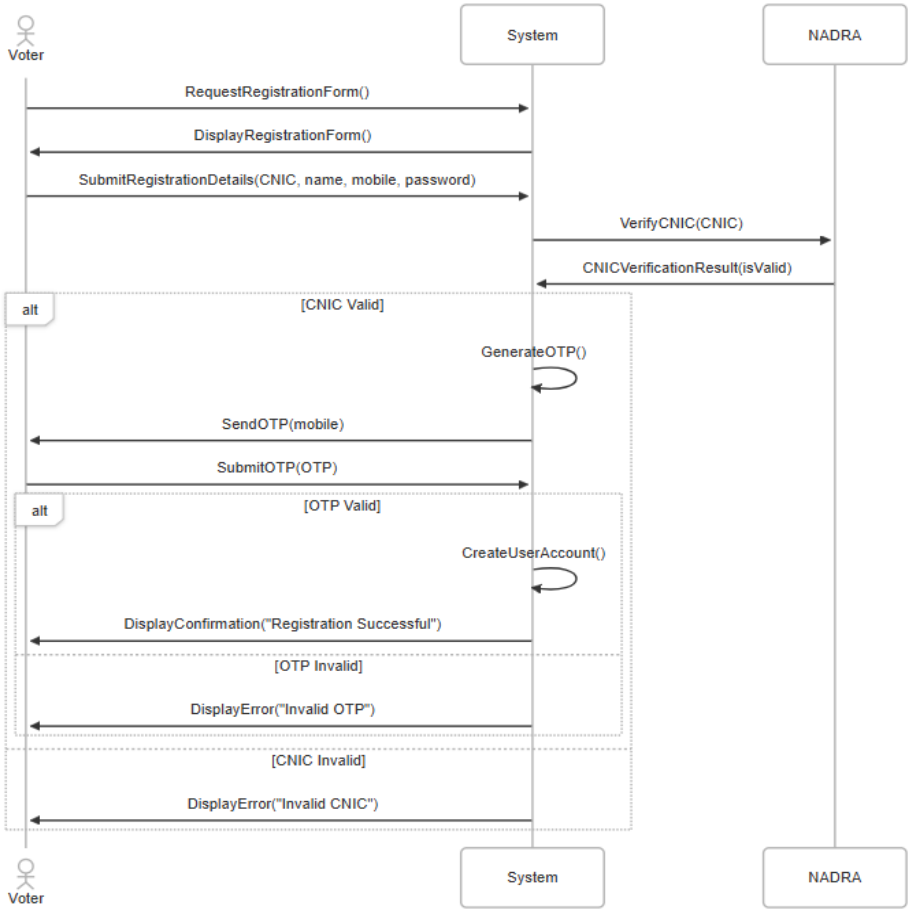
Step	User Action	System Response
1	Selects "Cast Vote"	Displays list of candidates
2	Chooses a candidate	Confirms selection
3	Submits vote	Records vote, disables further voting

Domain Model



System Sequence Diagrams and Contracts

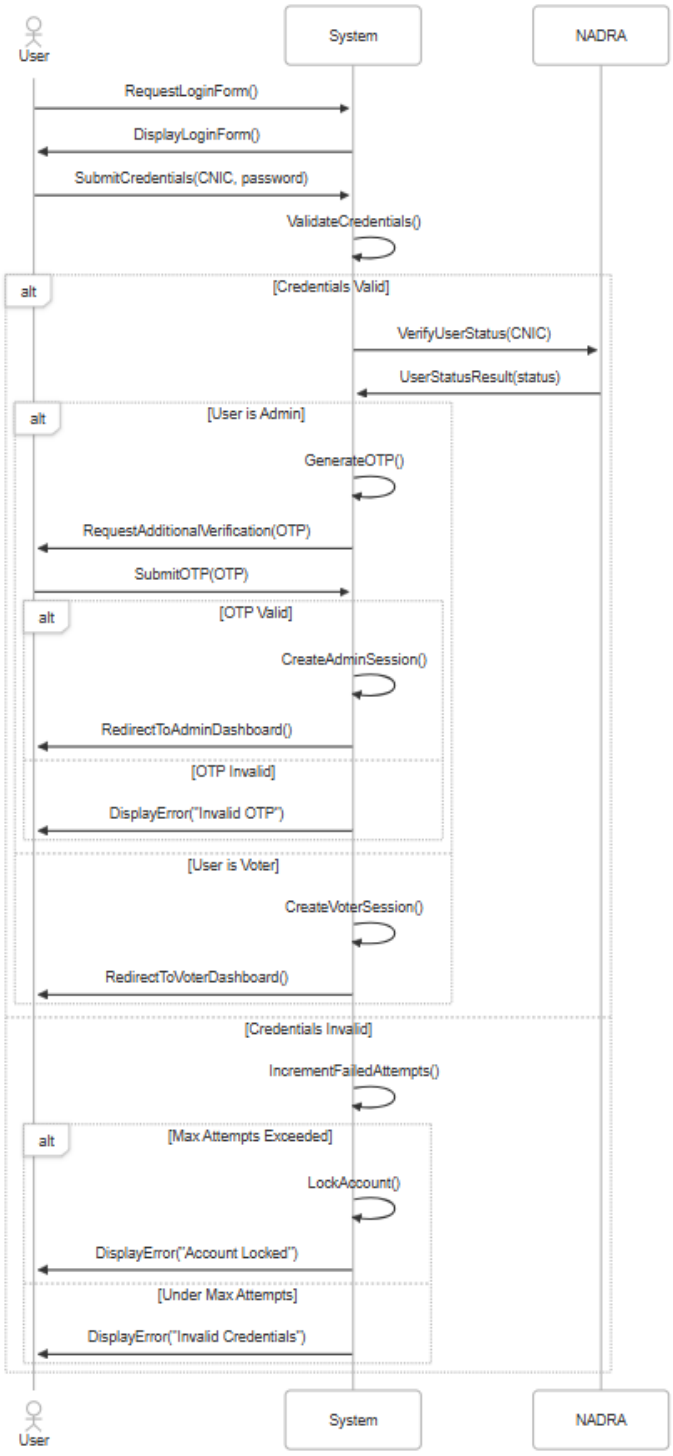
System Sequence Diagram UC-01:



Contract CO-1: Registration

Field	Description
Name	Registration
Responsibility	To register a new voter in the system
Type	Class level
Cross-Reference	UC-01
Notes	If the user is already registered, they will be directed to log in
Exceptions	• Error if OTP entered is incorrect • Error if user is already registered with provided CNIC • Error if CNIC validation fails against NADRA database
Output	User profile is created in the system
Pre-Conditions	None
Post-Conditions	The user has been registered and can now log in to the system

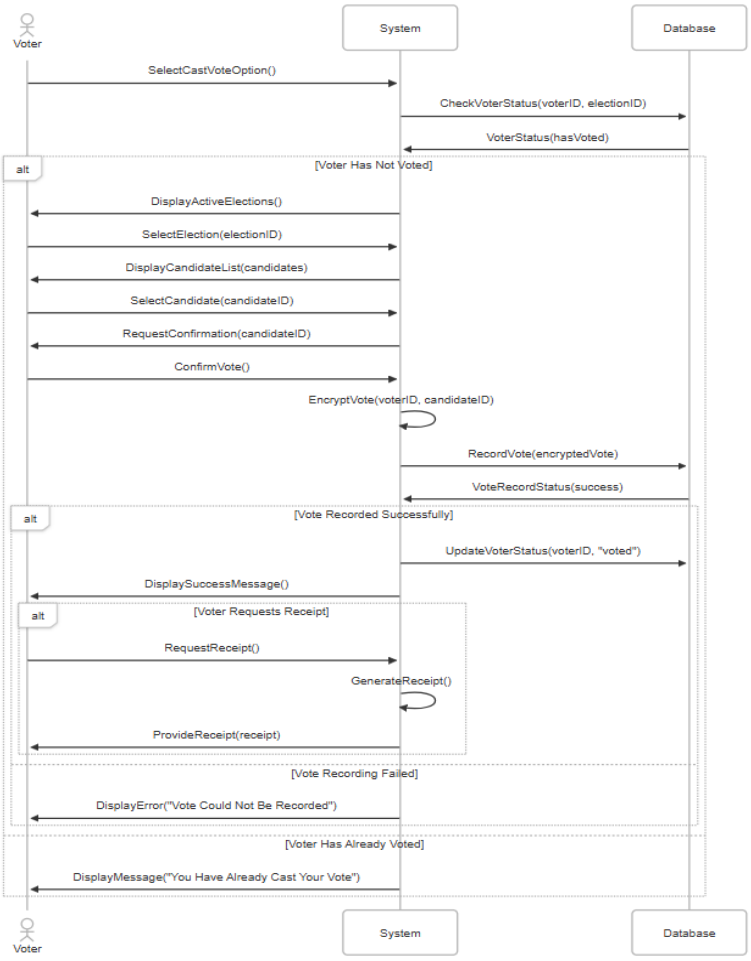
System Sequence Diagram UC-02:



Contract CO-2: Login

Field	Description
Name	Login
Responsibility	To authenticate a user and grant system access
Type	Class level
Cross-Reference	UC-02
Notes	Different user types (voter, admin) will be redirected to appropriate dashboards
Exceptions	• Error if credentials are invalid • Error if account is locked due to multiple failed attempts
Output	User session is created
Pre-Conditions	User must be registered in the system
Post-Conditions	User is logged in and redirected to appropriate dashboard based on role

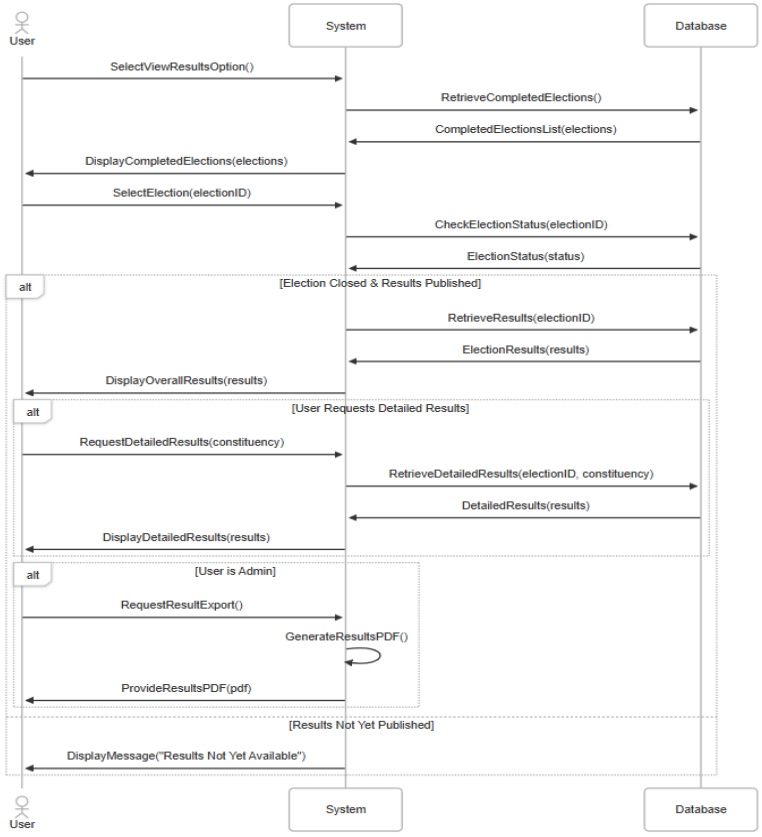
System Sequence Diagram UC-03:



Contract CO-3: Cast Vote

Field	Description
Name	Cast Vote
Responsibility	To record a voter's selection in an active election
Type	Class level
Cross-Reference	UC-03
Notes	Vote must be securely recorded and the voter's privacy maintained
Exceptions	• Error if voter has already cast a vote in this election • Error if election period is not active • Error if voter is not eligible for the specific election
Output	Vote is recorded securely and a digital receipt is generated
Pre-Conditions	• User must be logged in • User must be an eligible voter • Election must be active
Post-Conditions	• Vote is recorded in the system • Voter status is updated to "voted" • Digital receipt is provided to voter

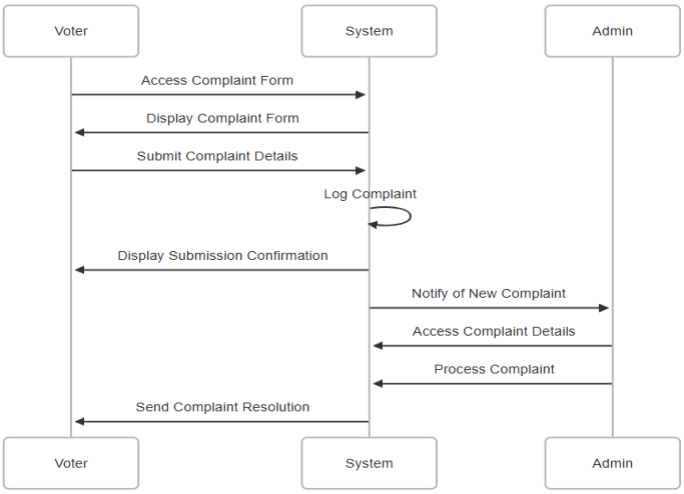
System Sequence Diagram UC-04:



Contract CO-4: View Election Results

Field	Description
Name	View Election Results
Responsibility	To display election results to users
Type	Class level
Cross-Reference	UC-04
Notes	Results should be presented in an easy-to-understand format with options for detailed views
Exceptions	• Error if election is still ongoing • Error if results are not yet finalized
Output	Election results displayed by constituency and overall
Pre-Conditions	Election period must be concluded
Post-Conditions	None

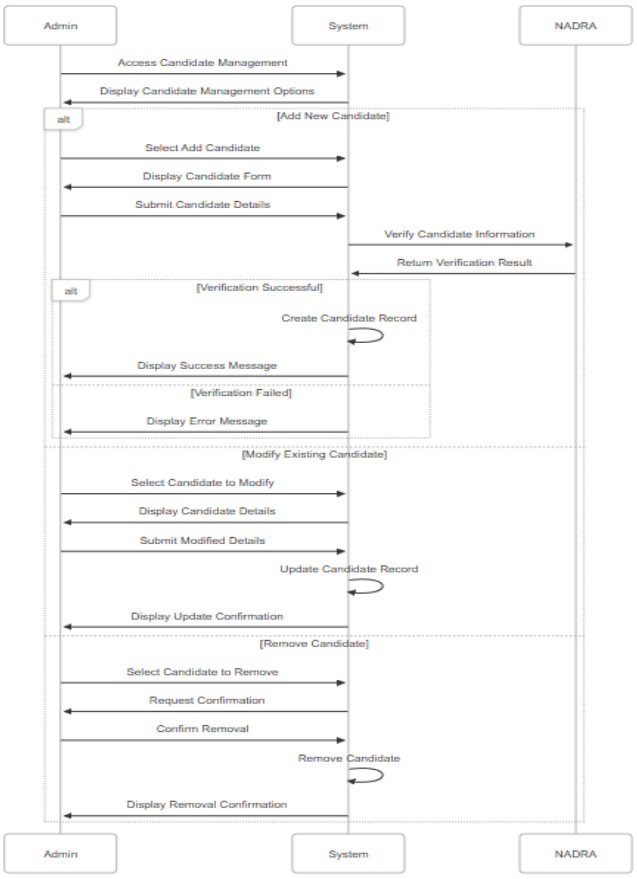
System Sequence Diagram UC-05:



Contract CO-5: Manage Election (Admin)

Field	Description
Name	Manage Election
Responsibility	To allow administrators to create and manage elections
Type	Class level
Cross-Reference	UC-05
Notes	Only authorized election officials can access this functionality
Exceptions	• Error if required election information is missing • Error if dates are invalid (e.g., end date before start date) • Error if constituencies or candidates are not properly defined
Output	Election is created, modified, or status changed
Pre-Conditions	User must be logged in with administrative privileges
Post-Conditions	• New election is created in the system OR • Existing election is modified OR • Election status is updated

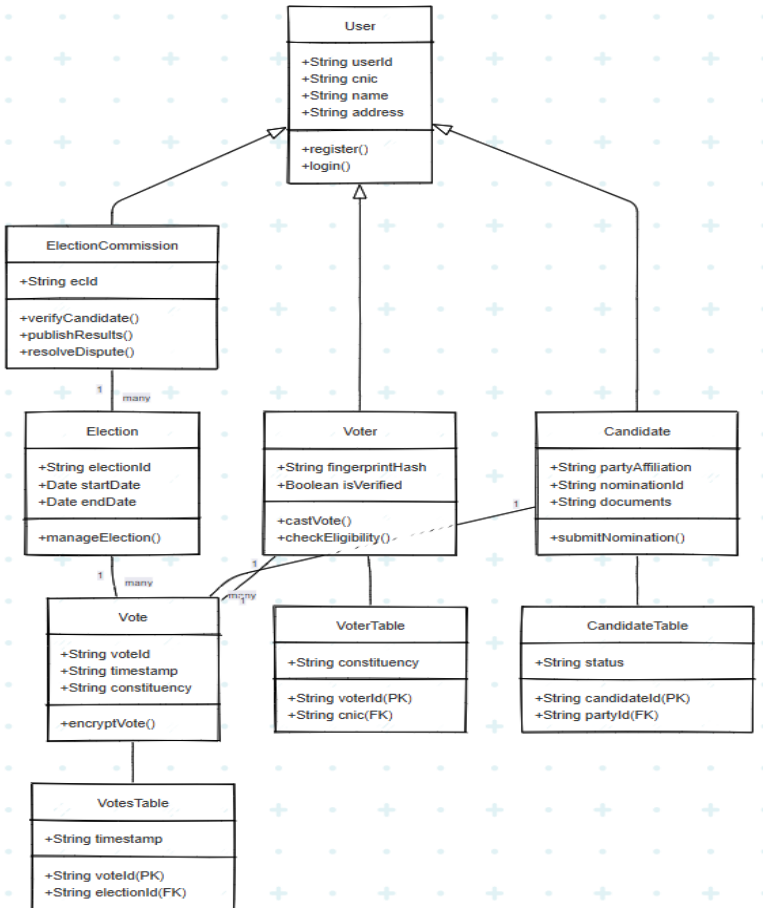
System Sequence Diagram UC-06:



Contract CO-6: Candidate Management (Admin)

Field	Description
Name	Candidate Management
Responsibility	To allow administrators to add, modify, and remove candidate information
Type	Class level
Cross-Reference	UC-CandidateManagement
Notes	Only authorized admin users can access candidate management functionality
Exceptions	• Verification failure from NADRA • Missing or incorrect candidate details • Attempt to modify or delete a non-existent candidate
Output	Candidate is added, modified, or removed successfully
Pre-Conditions	Admin must be logged in with appropriate privileges

Class Diagram:



Chapter 4

Results and Discussions

Test Cases

Table 1 System test cases and Outcomes

TC-ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status	Remarks
TC01	Voter Registration (Valid)	1. Enter valid CNIC, biometrics, and OTP. 2. Submit form.	Account created; confirmation SMS sent.	As Expected	PASS	NADRA integration successful.
TC02	Voter Registration (Invalid CNIC)	1. Enter invalid CNIC. 2. Submit form.	Error: "Invalid CNIC. Contact NADRA."	As Expected	PASS	Input validation works.
TC03	Candidate Nomination (Approved)	1. Submit complete documents. 2. EC verifies.	Status: "Approved" in dashboard.	As Expected	PASS	Automated document checks added.
TC04	Candidate Nomination (Rejected)	1. Submit incomplete documents. 2. EC reviews.	Status: "Rejected: Missing Documents."	As Expected	PASS	Email notification implemented.
TC05	Cast Vote (Authenticated)	1. Login via biometrics. 2. Select candidate. 3. Confirm.	Vote recorded; receipt generated.	As Expected	PASS	Encryption SHA-256 verified.
TC06	Cast Vote (Duplicate Attempt)	1. Try to vote again after successful submission.	Error: "You have already voted."	As Expected	PASS	Session blocking functional.
TC07	View Results (Public)	1. Access results page. 2. Filter by constituency.	Real-time results displayed with % progress.	As Expected	PASS	Data visualization optimized.
TC08	View Results (Admin)	1. EC logs in. 2. Download certified results PDF.	PDF with digital signature generated.	As Expected	PASS	Audit logs maintained.
TC09	Internet Disconnection Handling	1. Submit form during network failure. 2. Reconnect.	Data auto-saved; "Continue?" prompt shown.	As Expected	PASS	Local storage used effectively.

5. Project Closure

This project aimed to design a conceptual model for a secure, efficient, and accessible Online Voting System for Pakistan, with special focus on scalability, transparency, and legal adaptability. The documentation reflects theoretical application of software engineering principles using the Spiral Model to address the critical requirements of a national-level voting platform.

5.1 Key Outcomes

Although this system was not physically implemented or deployed, the following outcomes were defined and modeled as part of the documentation and design analysis:

- **Security Architecture:**
  - Conceptual prevention strategies for duplicate voting and unauthorized access were incorporated, including biometric authentication and CNIC-based validation modules.
- **System Performance (Simulated Estimates):**
  - Assumptions based on theoretical load testing suggest the model could handle over a million concurrent users with minimal latency under optimal infrastructure conditions.
- **User Interface and Usability:**
  - The prototype wireframes and UI flow were designed for simplicity and ease of navigation, with mock feedback indicating a user-friendly experience.
- **Transparency Features:**
  - Real-time result dashboards and audit logs were proposed to enhance system credibility among stakeholders.

5.2 Challenges and Design Resolutions

Challenge	Problem	Proposed Solution	Expected Outcome
NADRA Integration	Anticipated response delays	Use of local caching for recent verifications	Reduced processing time and dependency
Mobile Compatibility	UI text overflow on small screens	Responsive layout for small devices	Improved mobile usability and accessibility

These challenges were analyzed through **mock scenarios**, enabling effective design recommendations to be made.

5.3 Conceptual System Comparison

Feature	Proposed Online Voting System	Traditional EVMs	Paper Ballots
Vote Submission Time	~1.4 sec (theoretical model)	~2 min	~3 min
Cost per Vote (Estimate)	~$0.10	~$1.20	~$2.50
Accessibility	Mobile + Web	On-site only	On-site only

The model offers a cost-effective and accessible voting alternative in comparison to traditional methods, based on industry research and system modeling.

5.4 Future Recommendations

To further improve the theoretical design and its practical feasibility, the following enhancements are recommended:

- Incorporate **multilingual support** (e.g., Urdu, Pashto) for inclusive reach.
- Integrate **offline voting capability** for low-connectivity regions.
- Develop **AI-based fraud detection modules** to detect abnormal voting patterns.
- Explore blockchain-based vote integrity mechanisms for added security.

5.5 Project Value and Conclusion

This semester project presents a comprehensive and forward-thinking design for an Online Voting System in Pakistan. Although limited to documentation and simulation, the work demonstrates:

- A strong grasp of software process modeling using the Spiral Model.
- Thoughtful anticipation of security, usability, and legal challenges.
- A cost-effective and scalable approach to election modernization.

The project provides a valuable foundation for future research, prototyping, or academic exploration. With further development, it holds potential as a national digital voting solution tailored to Pakistan's democratic and technological landscape.