**E-Voting System**

**21BAI1861- THANGA SAI NAGA ANIRUDH  
21BAI1426- LOHITH KIRSHNA**

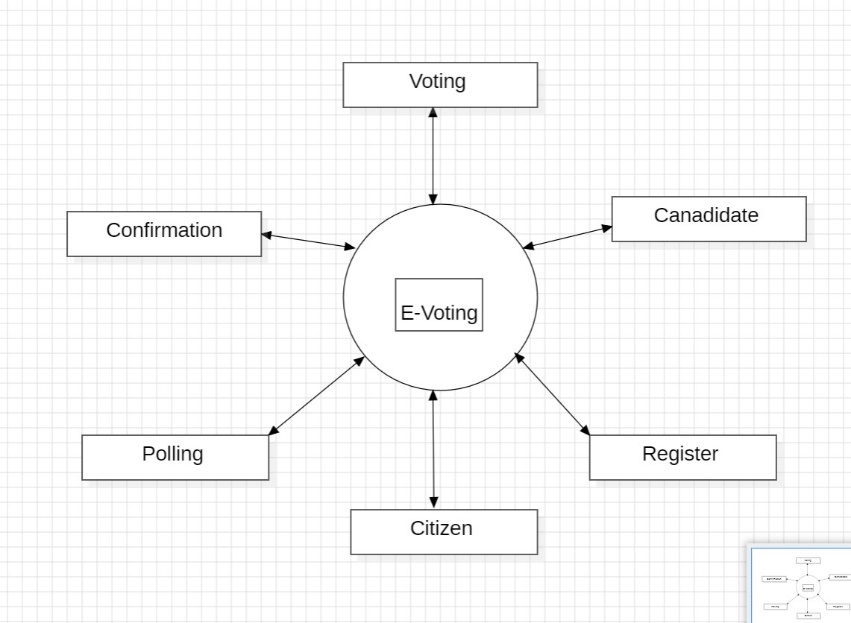
**Software Engineering**

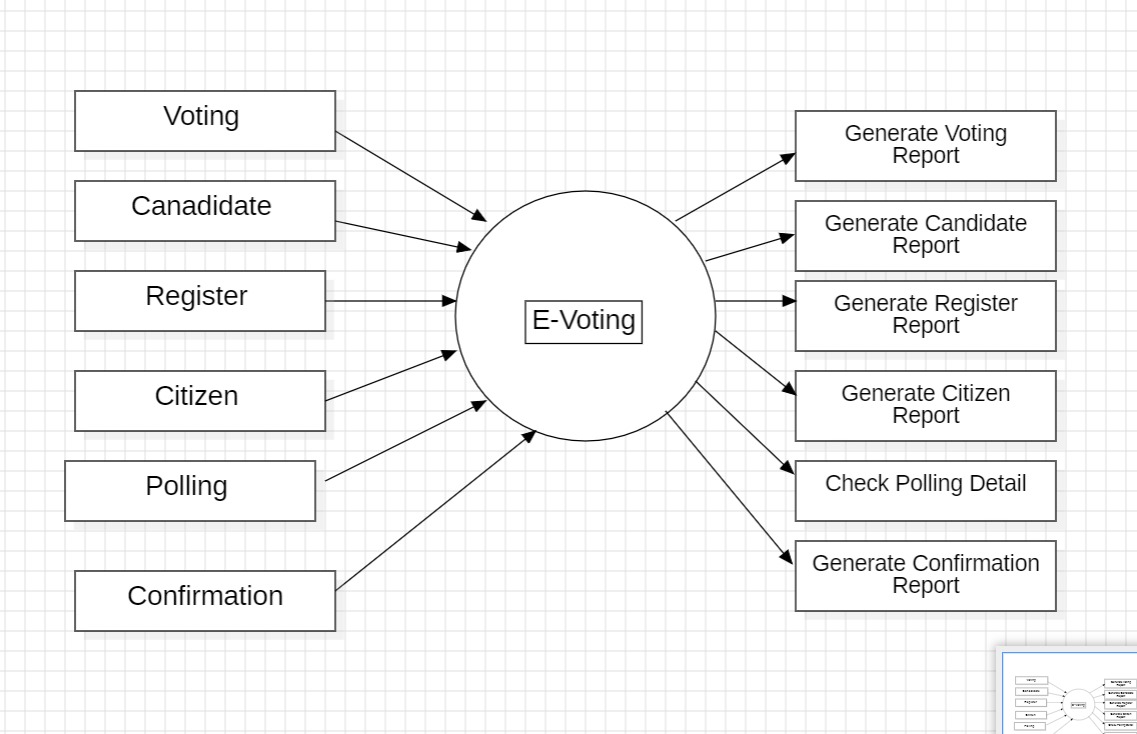
**Course code: BCE301P**

**Slot: C2 +TC2**

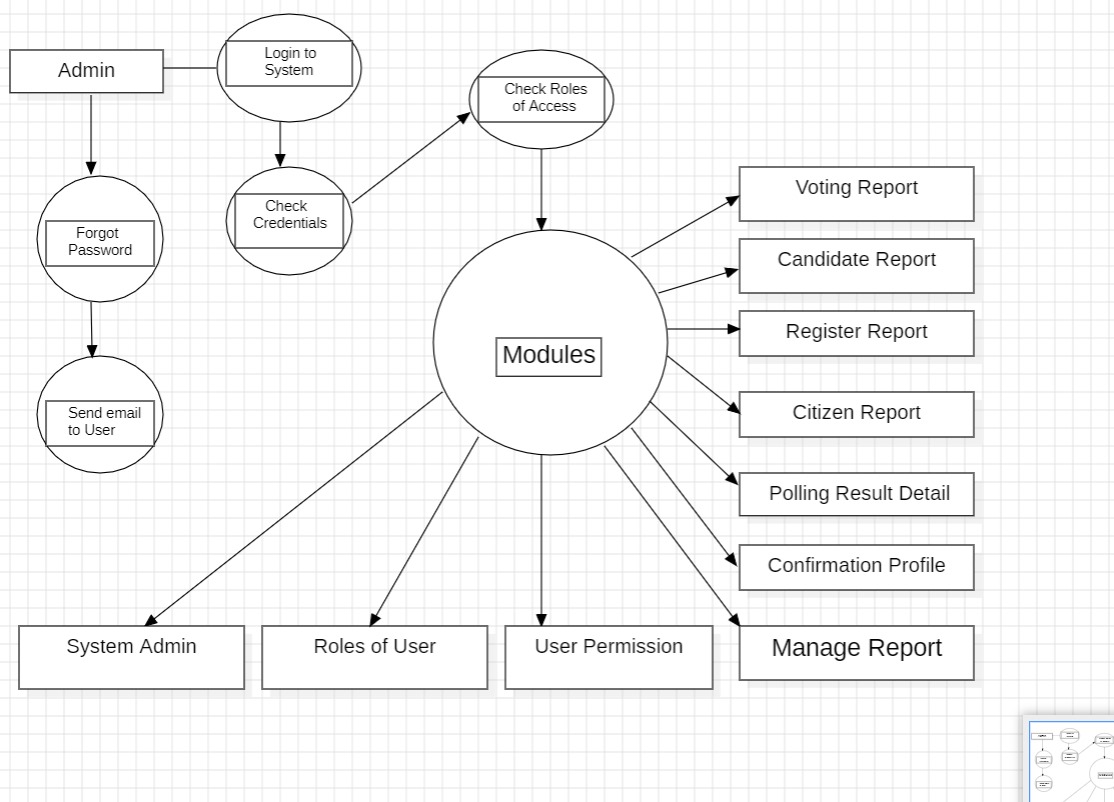
**Faculty: ILAKIYASELVAN N sir**

**Zero level dfd**

****

**First level dfd**

**Second level dfd**



**The e-voting process consists of several steps, as shown in the flowchart. The steps are as follows:**

**Register:** The first step in the e-voting process is registration. Citizens who want to participate in the voting process must register first.

**Confirmation:** After registering, citizens receive a confirmation message or email to verify their registration. This step ensures that only eligible citizens are registered for voting**.**

**Polling:** Once the registration is confirmed, citizens can participate in polling. Polling is the process of casting a vote for the preferred candidate.

**Voting:** After polling, the votes are collected and counted. This step is the actual voting process, where the votes are recorded and stored for further processing.

**E-Voting:** E-voting is the final step in the process. It is the electronic version of the voting process, where the votes are cast and counted using electronic systems.

The DFD includes entities like "Voter," "Administrator," "Candidate," "Ballot," "Vote," and "Database," representing different components and data flows within the e-voting system.

1. The "Voter" entity interacts with the system by logging in, viewing the ballot, casting votes, and viewing results.

2. The "Administrator" entity manages the system by overseeing voter registration, generating reports, and monitoring the voting process.

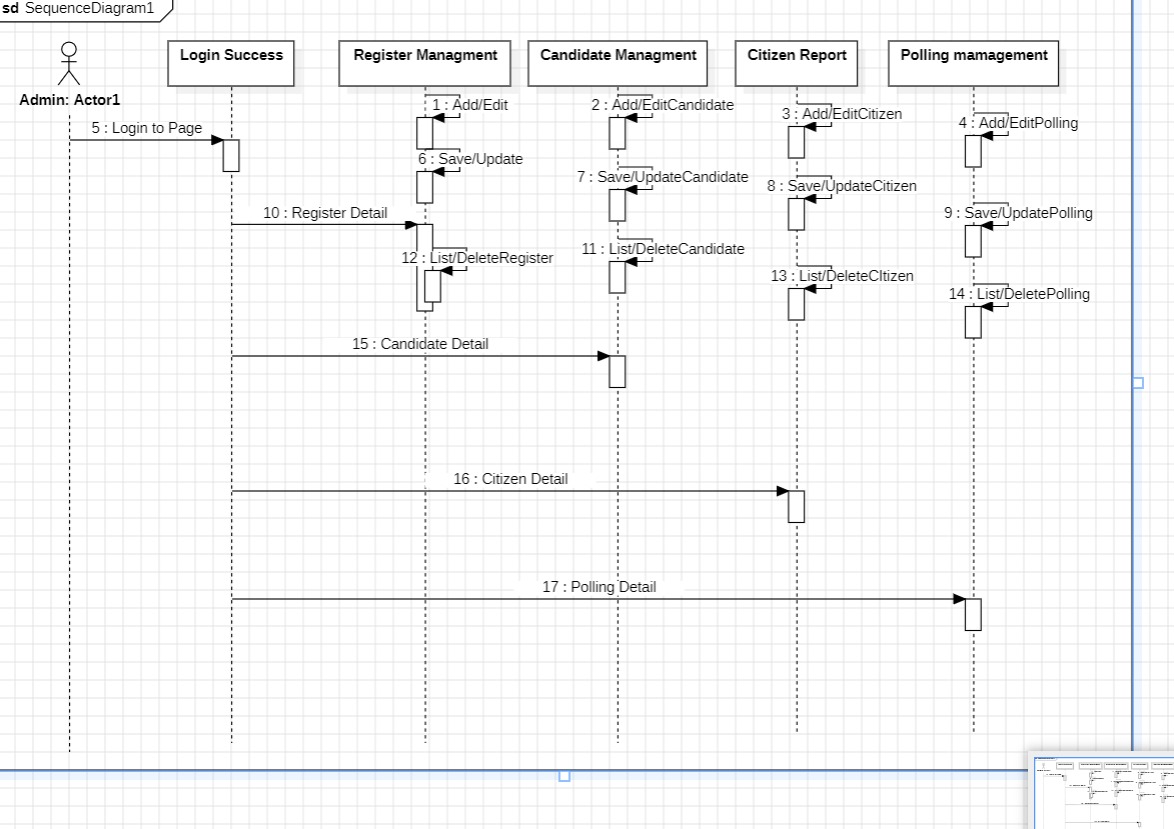
3. The "Candidate" entity represents individuals running for election, and their information is stored in the system.

4. The "Ballot" entity contains the list of candidates for voters to choose from during the voting process.

5. The "Vote" entity records the votes cast by voters and stores them in the system's database.

The data flows between these entities depict how information moves within the system, from the initial voter login to the final tallying of votes. This diagram provides a visual representation of the system's data flow and processes, aiding in understanding its functionality and interactions.

**Sequence diagram:**



The Sequence Diagram depicts the sequential flow of messages or actions between actors such as "Voter," "Administrator," and the "System" itself.

The "Voter" initiates the sequence by logging into the system.

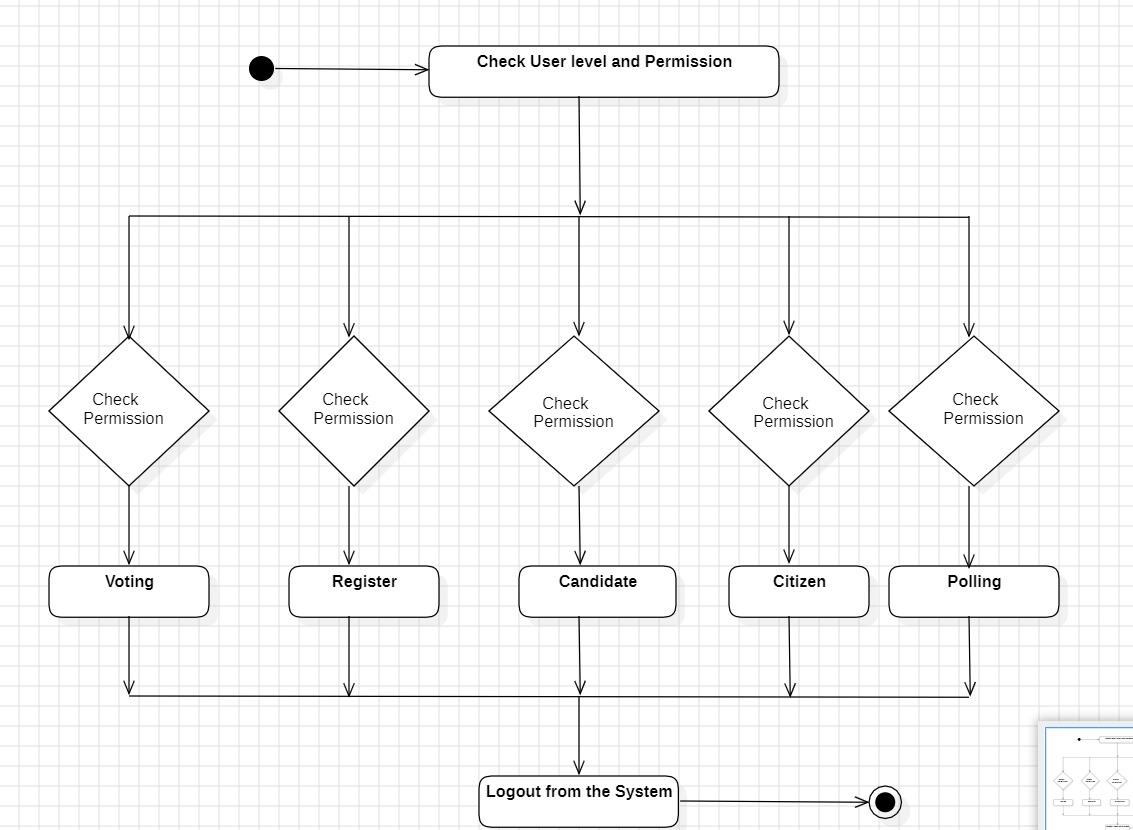
The "System" verifies the voter's credentials and presents the ballot.

The "Voter" selects a candidate and casts their vote.

The "System" records the vote and updates the database accordingly.

The "Administrator" monitors the voting process and generates reports as necessary.

Each message or action in the sequence diagram represents a step in the voting process, showcasing how different components interact and collaborate to facilitate the election procedure.

**activity diagram:** 

The Activity Diagram outlines the steps taken by voters and administrators within the e-voting system, from logging in to casting votes and generating reports.

1. Voters begin by logging into the system.

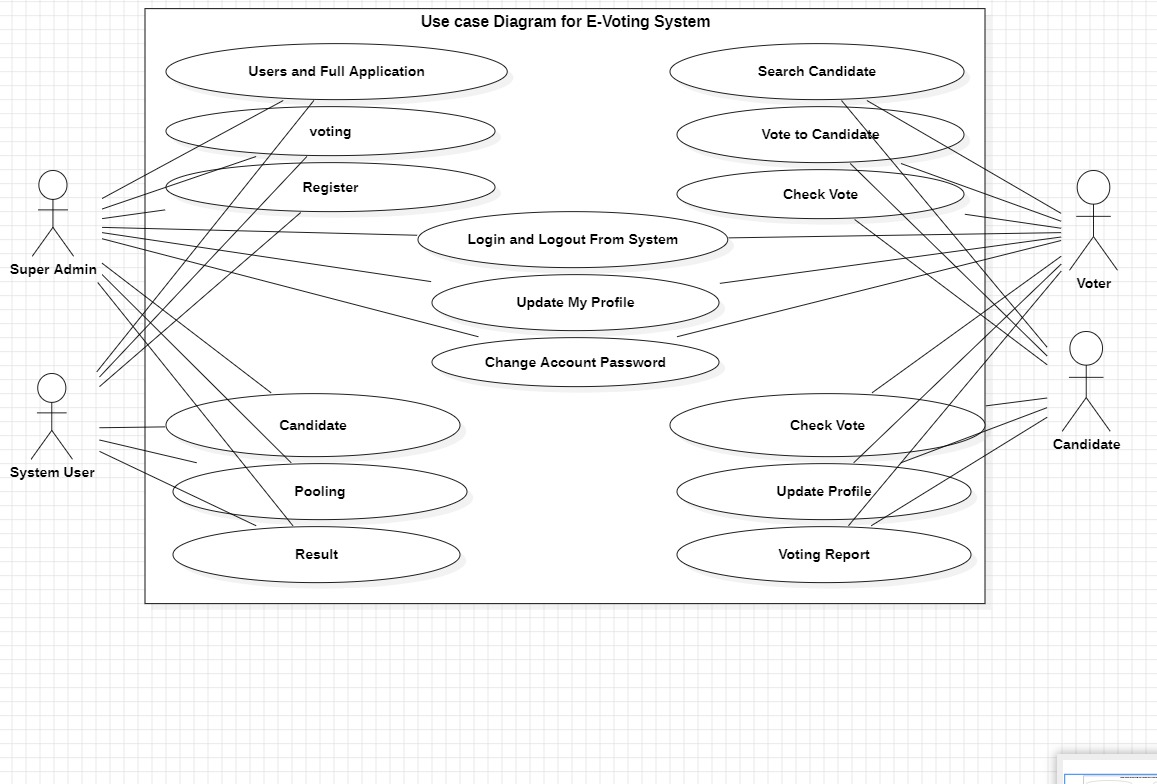
2. They then view the ballot and select their preferred candidates.

3. After making their choices, voters cast their votes.

4. Meanwhile, administrators manage tasks like voter registration, monitoring the voting process, and generating reports.

Each activity in the diagram represents a specific action or task performed within the e-voting system, providing a visual representation of the voting process flow.

**Use case diagram**



UML Use Case Diagram: A use case diagram is used to represent the dynamic behavior of a

system. It encapsulates the system's functionality by incorporating use cases, actors, and

their relationships. It models the tasks, services, and functions required by a

system/subsystem of an application. It depicts the high-level functionality of a system and

also tells how the user handles a system.

The UML Use Case Diagram for the e-voting system shows how users like voters and administrators interact with the system. Voters can log in, view the ballot, cast votes, and see results. Administrators manage voter registration, generate reports, and monitor the voting process. It's a visual map of who does what in the e-voting system.

THE DIFFERENCE IS ADDITION OF <<INCLUDE>> RELATION AT THE REQUIRED PLACES

the use case of "voting" includes the use cases of "vote verification" and "registration", which means that in order

to vote, a voter must first verify their vote and register. By

including these use cases, the details of the verification and registration processes do not need to be repeated in

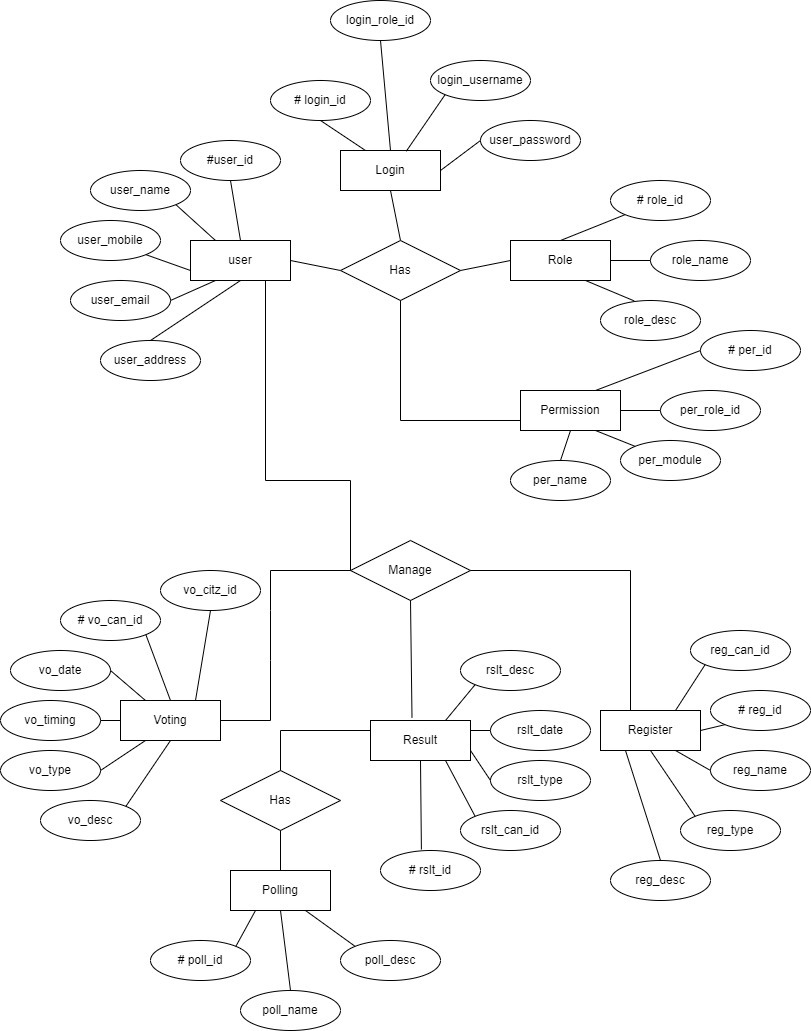
the voting use case description, and can be found in the

individual use case descriptions.

the use case of "election setup" includes the use case of "add voters or officers", meaning that in order to set up an

election, election officials must add voters or officers to the system. This allows for the details of adding voters or

officers to be described in a single use case, which can then be included in other use cases as needed.

**er diagram**   
  
  


Person: This entity represents an individual and has attributes such as SSN,

name, date of birth, place of birth, gender, marital status, address, job, and work

place.

2. Candidate: This entity represents a person who is running for a position and has

attributes such as being referred (0 or 1) and a list ID (foreign key).

3. Candidate List: This entity represents a list of candidates and has attributes such

as List ID, name, description, and logo.

4. Voter: This entity represents an individual who is eligible to vote and has

attributes such as being registered (1 or n) and a vote (1 or n).

5. Voter Vote: This entity represents an individual's vote and has attributes such as

ID, name, description, start date, and start time.

6. Vote Form: This entity represents a form that is used to cast a vote and has

attributes such as Form ID, name, description, choices count, choice type, end

date, and end time.

Relationships:

1. Person to Candidate: A Person can be a Candidate, indicating a one-to-many

relationship between Person and Candidate.

2. Candidate to Candidate List: A Candidate can belong to a Candidate List,

indicating a many-to-one relationship between Candidate and Candidate List.

3. Person to Voter: A Person can be a Voter, indicating a one-to-many relationship

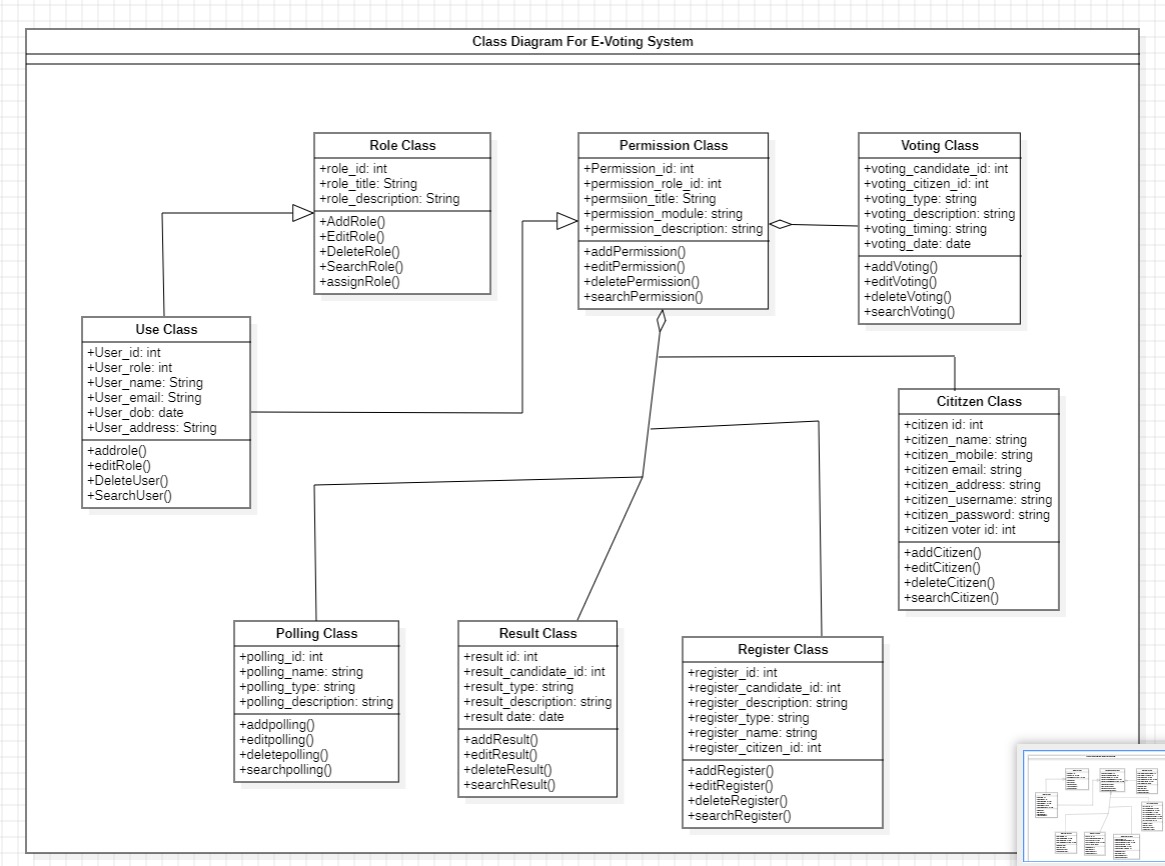
between Person and Voter.

4. Voter to Voter Vote: A Voter can have multiple Voter Votes, indicating a

one-to-many relationship between Voter and Voter Vote.

5. Voter Vote to Vote Form: A Voter Vote can have multiple Vote Forms, indicating a

one-to-many relationship between Voter Vote and Vote Form.

**Class diagram** 

The class diagram includes classes like "Voter," "Candidate," "Election," "Ballot," "Vote," and "Administrator," each representing different entities in the e-voting system.

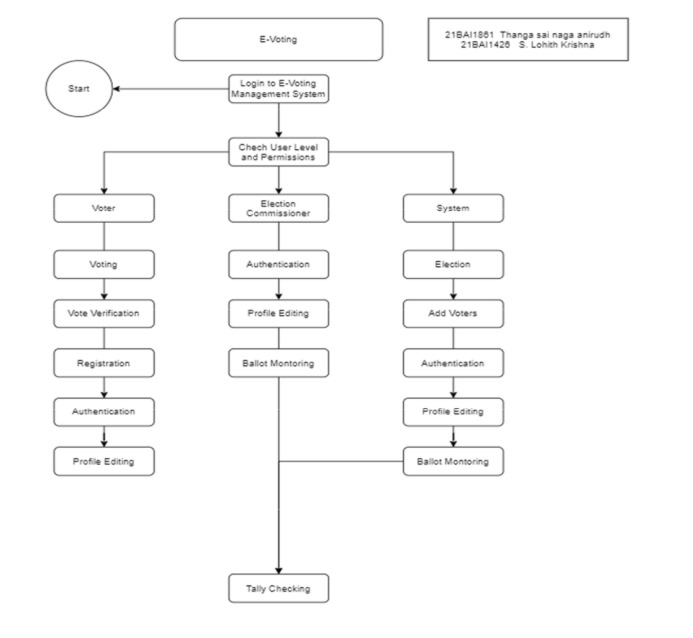
The "Voter" class stores information about voters, such as their ID, name, and login credentials.

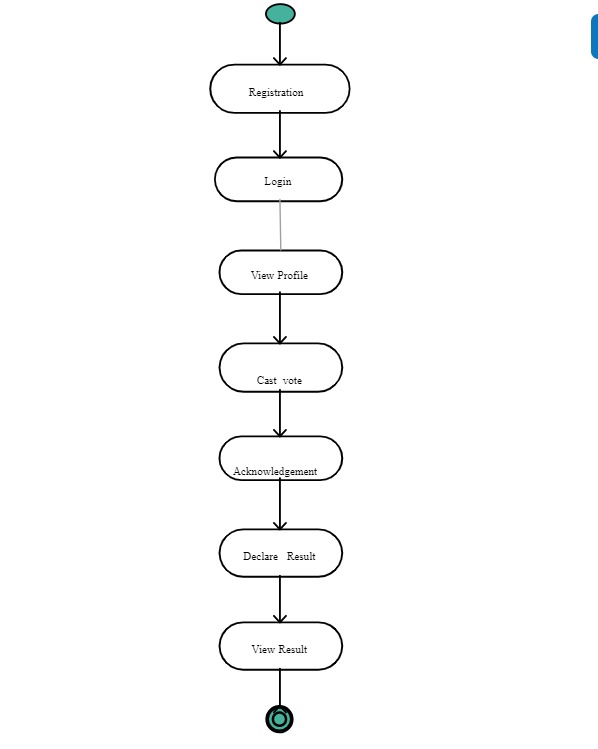
The "Candidate" class holds details about candidates running in an election, like their name and party affiliation.

The "Election" class manages the overall election process, including start and end dates.

The "Ballot" class represents the ballot paper containing the list of candidates for voters to choose from.

The "Vote" class records each vote cast by a voter, linking them to the chosen candidate.

The "Administrator" class handles administrative tasks, such as managing voter registration and monitoring the voting process.  
  
  
**WBS**

WBS e-voting, or Web-Based Secure Electronic Voting, is an online voting system that allows eligible voters to cast their ballots remotely using internet-enabled devices such as computers or smartphones. It provides a convenient and accessible method for voters to participate in elections without physically visiting polling stations. WBS e-voting systems typically incorporate security measures to ensure the integrity, confidentiality, and authenticity of the voting process, such as encryption, authentication protocols, and audit trails. These systems aim to enhance voter turnout, streamline the voting process, and improve overall election efficiency while maintaining the trust and credibility of the electoral process.  
  
 **STATE DIAGRAM**

Initialization State:

This is the starting state where the system is initialized but not yet ready for voting.

Voter Authentication State:

Upon initialization, the system prompts voters to authenticate themselves using a secure login

method, such as username and password or biometric authentication.

Ballot Selection State:

After successful authentication, voters enter this state where they can select the ballot or the

election they want to participate in.

Voting State:

Once the voter selects the ballot, they enter the voting state where they can cast their votes for

the candidates or options presented.

Confirmation State:

After completing the voting process, the system prompts the voter to confirm their selections

before finalizing the vote.

Vote Submission State:

Upon confirmation, the vote is submitted securely to the system.

Vote Counting State:

After the voting period ends, the system enters the vote counting state where it tallies the votes

and calculates the results.

Result Display State:

Once the vote counting process is complete, the system displays the election results to

authorized personnel or the public, depending on the system's design.

End State:

After the results are displayed, the system transitions to the end state, where it awaits further

instructions or shuts down until the next election cycle.