Report On Election Management System

Group 3

Abstract:

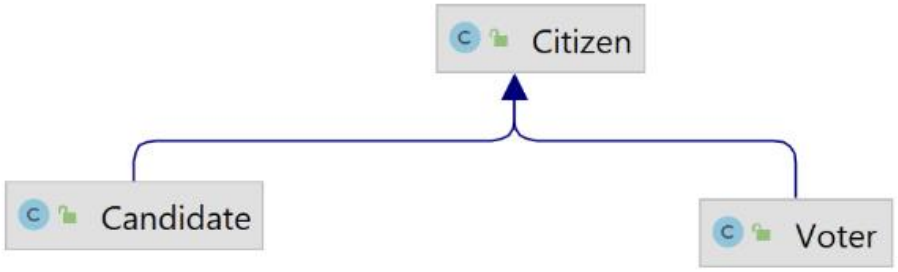
This Project is designed to automate the overall election system of India.This project can be used to conduct the elections efficiently.This project deals with the design and development of an election system using Aadhar authentication, Overall process of election would be handled by this project including Citizen Registration,Aadhar Card generation,Voter Registration,Candidate addition,release of Candidate list for each Constituency, Vote Casting, Vote counting and declaration of results in various formats.

Problem Definition:

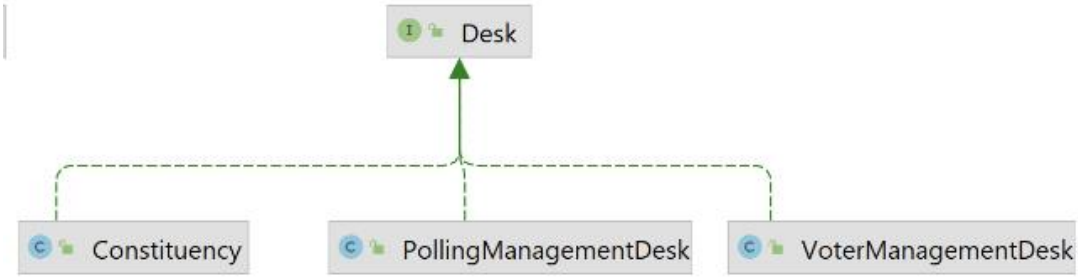
The main Goal of our project is to make the process of elections efficient and also work towards increasing voter turnout. This system handles all the important functions in an election such as registration, authentication, voting process, counting and finally displaying the result, the number of personnel required to conduct elections can be reduced to only a small absolutely essential number.In the present system a voter can vote only at the given constituency we can solve this problem by allowing voters to cast their vote from any polling station, and this system solves this issue as we extract all the details required like the constituency using the Aadhar number. Another hurdle while conducting elections is the problem of impersonation during voting,Our system doesn't allow the voter to vote more than once and there is no need for any human involvement to prevent such cases.

Methodology:

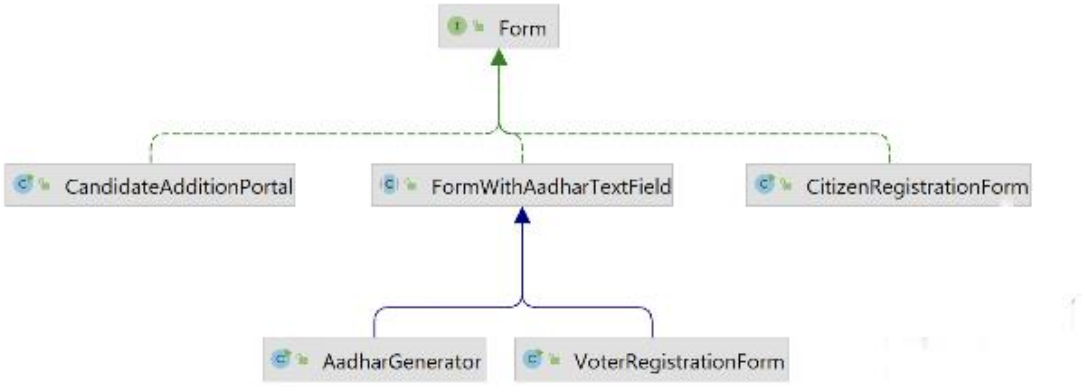
We begin with representing each citizen within a constituency as an object. It can be observed that each citizen may or may not be eligible for voting and may or may not contest in the elections as a candidate. Hence, a voter/candidate is a citizen with additional functions. With this basis, Voter and Candidate classes were extended from citizen class. An aadhar ID is assigned to each of the registered citizens via which they can access processes like Voter Registration, AadharCard generation etc.



Each Constituency in an election performs function with regard to voter management and polling management. Function related voter management include registration of voter, addition of voters into the database, defining criteria for voter eligibility, checking if the voter has already registered himself as a citizen of the constituency and computing voter turnout with other relevant data likewise . Similar tasks are performed in the area of Polling management relevant with respect to candidates like keeping track of votes, candidates nomination, declaring the winner etc. With this basis, classes VoterManagementDesk and PollingManagementDesk were created with the Constituency class having an object each of both. It was also observed that each of the above classes share similar features such as registering individuals onto the database, accessing citizen data with aadhar, offering a registration portal, checking if the aadharNumber is valid. Hence, classes VoterManagementDesk, PollingManagementDesk and Constituency implement a Desk.java interface.



The Form.java interface outlines the features of a registration/addition form which collect data, check if the same data is in the ideal format and cater the same data into relevant objects so that necessary function is performed and display appropriate return message. This interface is implemented by CitizenRegistrationForm, VoterRegistrationForm, AadharGenerator and CandidateAdditionPortal. The CitizenRegistrationForm collects data through the UI and calls the registerIndividual() method of Constituency class which then appends the data into the text files. The AadharID generated will have the citizen data encoded within. VoterRegistrationForm and AadharGenerator forms do the same but call methods registerIndivual() and generateAadharCard() from VoterMangementDesk respectively. CandidateAdditionPortal calls relevant methods from PollingManagementDesk once data in correct format is collected. The data generated if any is stored in simple text files.

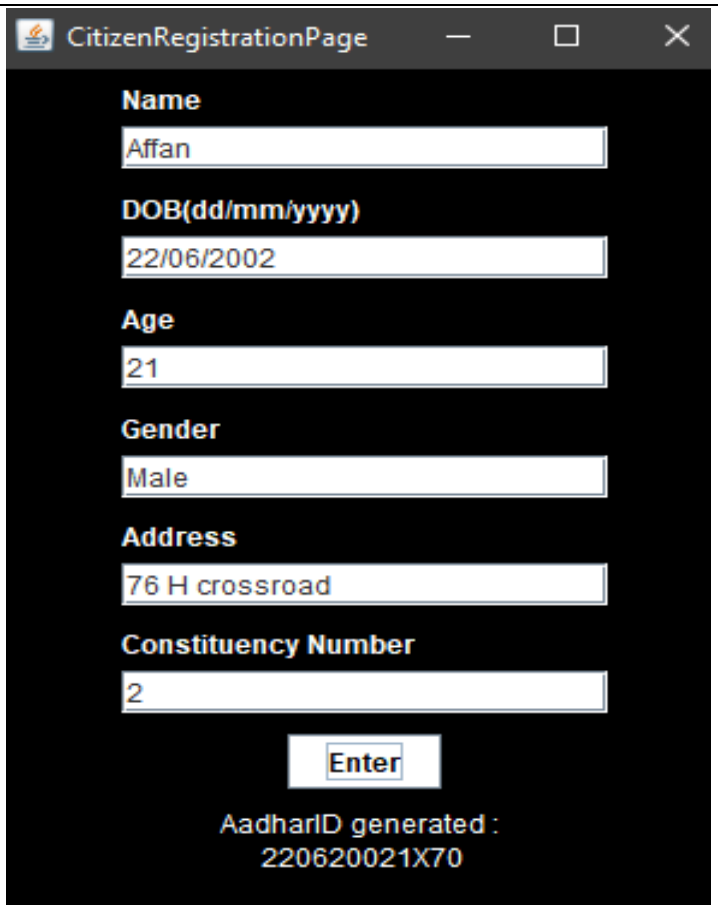


The Party.java class keeps tracks of all the candidates it has. A party is registered as soon as a candidate with that party name is added into the candidateList by the admin through CandidateAdditionPortal. The list of PartyNames Contesting in the election will be added to the Parties.txt file.

For the process of voting, PollingBooth has been implemented. The PollingBooth has an ArrayList of Constituencies as an instance variable so that access to voters and candidates data is possible. The registerVote() method of this object simulates the process of voting and asks for aadharID from the user and displays candidates contesting in the constituency corresponding to the AadharID. Once the vote has been cast, the voter will be marked voted to prevent him from voting again. The cycle continues and votes are updated alongside with the updateVote() method in pollingManagementDesk of the corresponding constituency. Once the election day is done with, the PollingBooth object can be passed onto the constructor of Result class. The Result class has necessary methods to compute the winning Party, retrieving data from the constituencies. The admin can then release the result as a series of PDF documents each containing results in a variety of formats. The packages com.management.populace and com.management are made so as to provide encapsulation in a way that prevents citizens and candidates from accessing the protected methods of com.management.

Result:

1. CitizenRegistrationForm:
The form asks for Name, Age, DOB, Gender, Address and Constituency number of the user. The Swing UI form checks if the data entered is in correct format and appends the citizen data into text files corresponding each constituency. The correct data will have date in DD/MM/YYYY format with address containing the door number at the start. An error saying the data entered is not in correct format will be shown if so. The Aadhar ID generated will be displayed. The citizen, once registered, can now access other features such Voter Registration, Aadhar Card generation and Vote Casting with his AadharID.

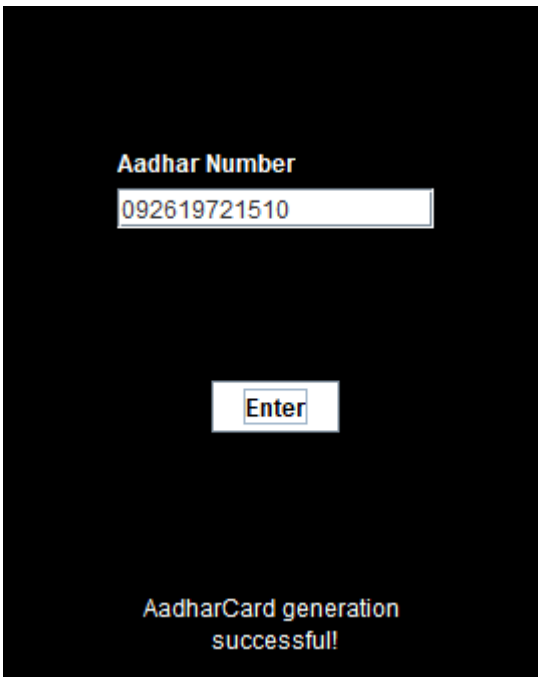


2. VoterRegistrationForm:

A citizen who has registered himself as a citizen can now register himself as a Voter if his age is greater than 18. The Swing UI form corresponding to this feature will check if the Aadhar ID entered into the form is valid, if the actor has already registered himself as a citizen, if the corresponding citizen is eligible for voting and append Voter into the Voter-list accordingly. If the aadhar ID entered is not valid, and if the aadhar number contains invalid characters, an error saying the same will be shown on the textPane present towards the bottom. The isDataInformat() makes sure the entered aadhar ID is in right format before calling VoterManagementDesk to fetch corresponding data.
Note: Valid characters in an aadharID include numbers from 0 to 9 and letter ‘X’ which represents 10.

3. AadharGenerator:

A citizen who registered himself can access this feature and generate his/her Aadhar card in PDF format by merely entering their AadharID. Mechanism to fetch citizen data corresponding to the AadharID entered has been implemented. If the aadharID entered is not valid, and if the aadhar number entered contains invalid characters, an error saying the same will be shown on the textPane present towards the bottom.



4. Candidate Addition:

The management/admin can add candidates into the candidate list via this feature. The Swing UI form corresponding to this feature will check if the Aadhar ID entered into the form is valid, if the actor has already registered himself as a citizen and append Candidate data into the Candidate-list accordingly.

5. Release of Candidates List and Parties Contesting:

Once the candidates have been registered into the system, the management/admin can release the candidate list for each constituency in pdf format. This will include display of Political parties contesting in the elections and their CM face.

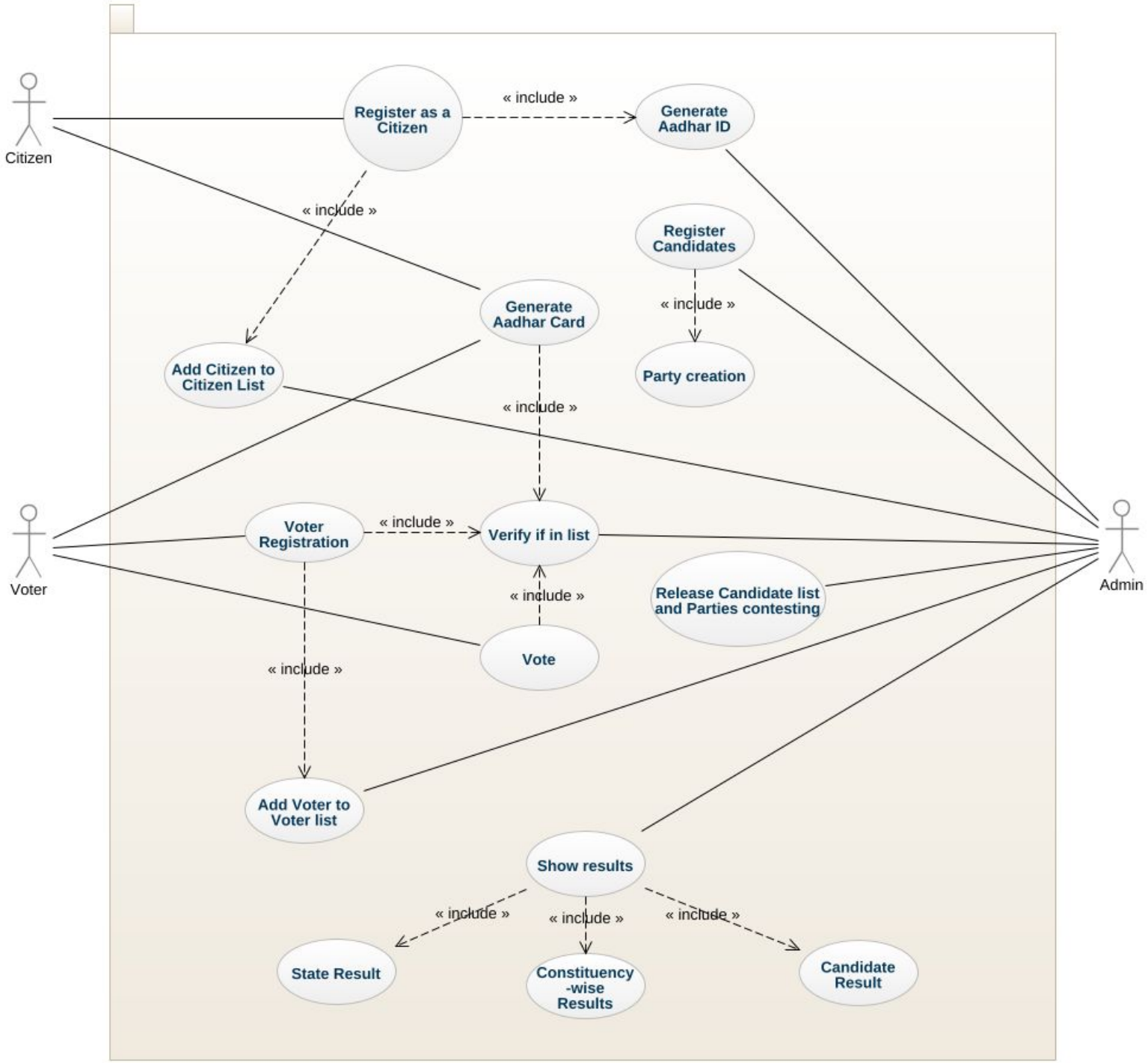
6. Vote Casting:

Registered voter can enter his/her AadharID and vote via this feature. The system checks if the voter has voted and proceeds only if the voter has not voted before. Constituency corresponding to the voter is retrieved from the AadharID and candidates contesting in the corresponding constituency are shown to the voter. Voters can then cast their vote. Once done with voting, the voter will be marked voted to prevent him from voting again.After each cast of vote, termination key will be asked which will be known only to the admin. Any other voter can enter an int and xontinue with voting likewise. Once termination key is entered by the admin, the entire result will be declared automatically in PDF.

7. Result declaration:

The admin.management can then release results corresponding to the election with relevant data such Winning Candidates, Winning Party, Voter turnout etc.

Features numbered 1-5 are accessible through PreElection.java and the rest are accessible through ElectionAndResult.java. Details on compile and run are provided in README.md file.


Citizen
Register as a Citizen
« include »
Generate Aadhar ID
« include »
Register Candidates
Generate Aadhar Card
« include »
Party creation
Add Citizen to Citizen List
« include »
Voter
Voter Registration
« include »
Verify if in list
Admin
« include »
Release Candidate list and Parties contesting
Vote
« include »
« include »
Add Voter to Voter list
Show results
« include »
« include »
« include »
State Result
Constituency -wise Results
Candidate Result