E-VOTING SYSTEM

FUNCTIONAL REQUIREMENT DOCUMENT

INTROUCTION

An electronic voting system, or e-voting, is a technology-driven approach to the casting and counting of votes in elections. It employs electronic means, such as online platforms or electronic voting machines, to enhance efficiency, accuracy, and accessibility in the voting process. E-voting reduces the reliance on paper, speeds up result announcements, and often incorporates security measures like encryption and auditability. While offering convenience and environmental benefits, concerns about cybersecurity and system vulnerabilities have prompted ongoing debates about the overall reliability and security of e-voting systems.

1- Features of our system are:

1. Verification of admin.
2. Add entries in NADRA file.
3. Update NADRA file.
4. Delete entries in NADRA file.
5. Display entries in NADRA file.
6. Add entries in Election Commission file.
7. Update Election Commission file.
8. Delete entries in Election Commission file.
9. Display entries in Election Commission file.
10. Taking information from candidate.
11. Verifying entered information from Election Commission database.
12. Register the candidate in Candidate file if it is verified from Election Commission.
13. Don’t allow candidate to register if already registered.
14. Create candidate login.
15. Candidate can view his votes through his login.
16. Taking information from voter.
17. Verifying entered information from NADRA database.
18. Register the voter in Registration file if it is verified from NADRA.
19. Don’t allow voter to register again if already registered.
20. Create voter login.
21. Display candidates of same province as entered by voter.
22. Voter can cast vote after registration or login.

23. Store vote in Vote Stored file.
24. Don’t allow user to cast vote again if already casted.
25. Allows voter to view his casted vote after login.
26. Count votes from vote stored file.
27. Display winner.
28. Display vote casted by a voter by taking his cnic.
29. Display number of votes of a candidate by taking his name.
30. Sort provinces on their number of votes.
31. Sort candidates on their number of votes.
32. Sort cities on their number of votes.
33. Sort parties on their positions.

2- There are total 3 actors in our system. They are:

- Admin
- Voter
- Candidate.

3- Actors can interact with system as following:

- Admin can manage databases of NADRA and Election Commission by logging in.
- Candidates can make their profiles by registration and see their secured votes.
- Voters can make their profiles by registration, cast vote to candidates of same city and see their vote details.
- Any actor of the system can see any detail of election.

Design

Modules:

There are 4 modules:

- Admin module
- Candidate module
- Voter module
- Display module

Functional Requirements:

Verification of Admin:

- The system must authenticate the admin by checking credentials against stored information.

NADRA File Operations:

- Add, update, delete, and display entries in the NADRA file.

Election Commission File Operations:

- Add, update, delete, and display entries in the Election Commission file.

Candidate Operations:

- Collect information from candidates.
- Verify candidate information from the Election Commission database.
- Register the candidate in the Candidate file if verified.
- Prevent candidate registration if already registered.
- Create a login for candidates.
- Candidates can view their votes through their login.

Voter Operations:

- Collect information from voters.
- Verify voter information from the NADRA database.
- Register the voter in the Registration file if verified.
- Prevent voter registration if already registered.
- Create a login for voters.
- Display candidates of the same province as entered by the voter.
- Allow voters to cast votes after registration or login.
- Store votes in the Vote Stored file.
- Prevent users from casting votes again if already casted.
- Allow voters to view their casted votes after login.

Vote Counting and Display:

- Count votes from the Vote Stored file.
- Display the winner.
- Display votes casted by a voter by taking their CNIC.
- Display the number of votes of a candidate by taking their name.
- Sort provinces, candidates, cities, and parties based on their number of votes or positions.

Non-functional Requirements:

- Security:
  The system should implement robust security measures for user authentication and data protection.

- Performance:
  The system must perform efficiently, especially during peak times of voter activity.

- Reliability:
  The system should be reliable, with minimal downtime and a low risk of errors.

- Scalability:
  The system should scale to handle a growing number of users and data.

- Auditability:
  Maintain an audit trail for all system interactions for transparency and accountability.

- Usability:
  Ensure a user-friendly interface for both admin, candidates, and voters.

- Maintainability:
  Design the system for easy maintenance, updates, and modifications.

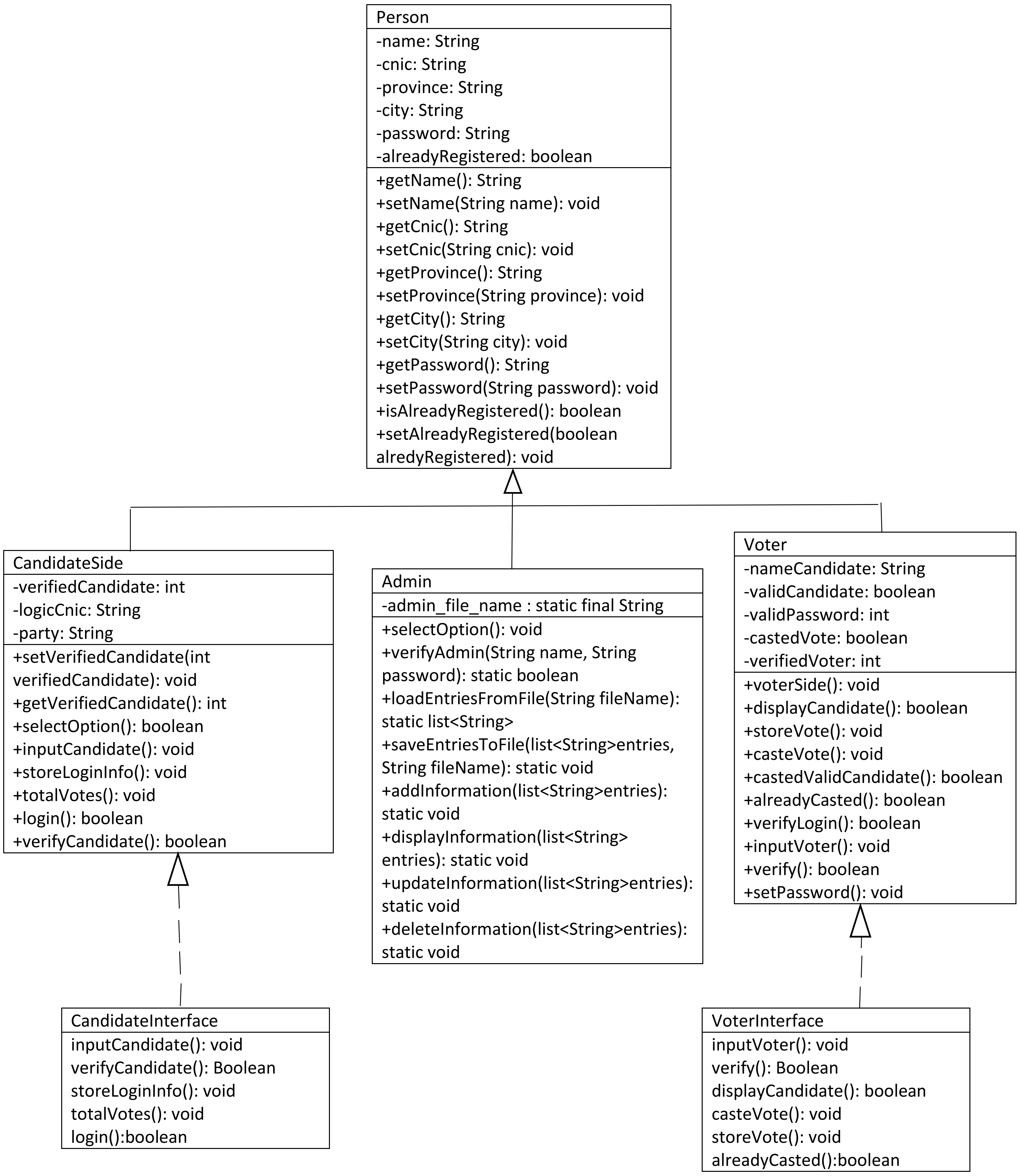
- Ethical Considerations:
  Adhere to ethical standards, ensuring fairness, transparency, and privacy.

- Environmental Impact:
  Minimize environmental impact, such as paper usage, where applicable.

Class Diagram:

For implementation of the system, we should have following classes:

- Person class
- Admin class
- CandidateSide class
- Voter class
- Display class
- EVoting class
- Candidate side interface
- Voter interface

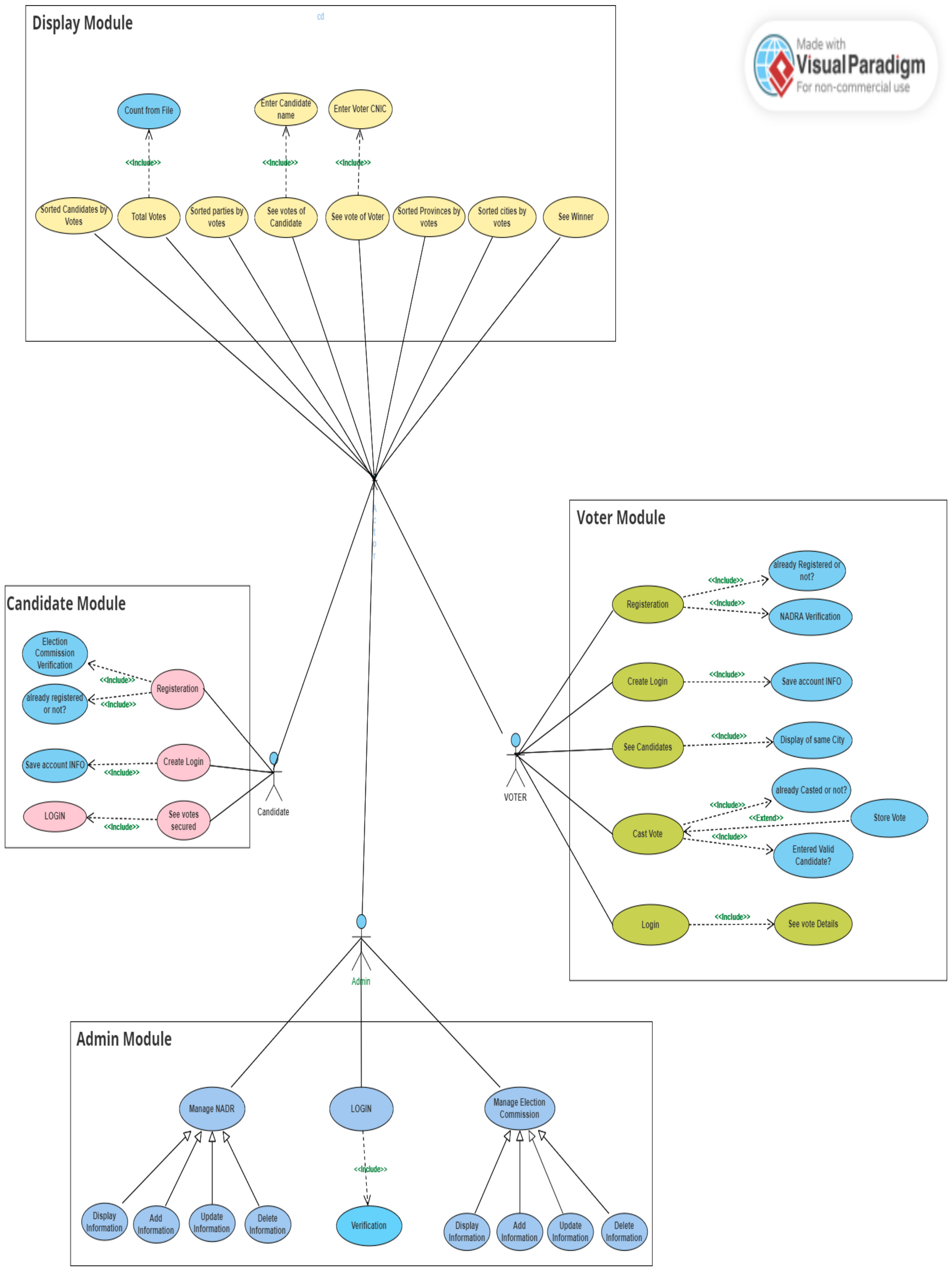

Person
-name: String
-cnic: String
-province: String
-city: String
-password: String
-alreadyRegistered: boolean
+getName(): String
+setName(String name): void
+getCnic(): String
+setCnic(String cnic): void
+getProvince(): String
+setProvince(String province): void
+getCity(): String
+setCity(String city): void
+getPassword(): String
+setPassword(String password): void
+isAlreadyRegistered(): boolean
+setAlreadyRegistered(boolean alredyRegistered): void
CandidateSide
-verifiedCandidate: int
-logicCnic: String
-party: String
+setVerifiedCandidate(int verifiedCandidate): void
+getVerifiedCandidate(): int
+selectOption(): boolean
+inputCandidate(): void
+storeLoginInfo(): void
+totalVotes(): void
+login(): boolean
+verifyCandidate(): boolean
Admin
-admin_file_name : static final String
+selectOption(): void
+verifyAdmin(String name, String password): static boolean
+loadEntriesFromFile(String fileName): static list<String>
+saveEntriesToFile(list<String>entries, String fileName): static void
+addInformation(list<String>entries): static void
+displayInformation(list<String> entries): static void
+updateInformation(list<String>entries): static void
+deleteInformation(list<String>entries): static void
Voter
-nameCandidate: String
-validCandidate: boolean
-validPassword: int
-castedVote: boolean
-verifiedVoter: int
+voterSide(): void
+displayCandidate(): boolean
+storeVote(): void
+casteVote(): void
+castedValidCandidate(): boolean
+alreadyCasted(): boolean
+verifyLogin(): boolean
+inputVoter(): void
+verify(): boolean
+setPassword(): void
CandidateInterface
inputCandidate(): void
verifyCandidate(): Boolean
storeLoginInfo(): void
totalVotes(): void
login():boolean
VoterInterface
inputVoter(): void
verify(): Boolean
displayCandidate(): boolean
casteVote(): void
storeVote(): void
alreadyCasted():boolean

Display
-vote_stored_file: static final String
+displayWinner():void +countVotes(String candidateName): int +displayVote(String voterCnic): void +displayVotesOfCandidate(String candidateName): void +sortProvince(): void +sortCandidates(): void +sortParty(): void +sortCity(): void

EVoting
+ main(String[] args): static void

Use Case Diagram:

Use case diagram of the system is as follows:


Display Module
cd
Count from File
Enter Candidate name
Enter Voter CNIC
<<Include>>
<<Include>>
<<Include>>
Sorted Candidates by Votes
Total Votes
Sorted parties by votes
See votes of Candidate
See vote of Voter
Sorted Provinces by votes
Sorted cities by votes
See Winner
Actor
Voter Module
already Registered or not?
<<Include>>
Registeration
<<Include>>
NADRA Verification
Create Login
<<Include>>
Save account INFO
See Candidates
<<Include>>
Display of same City
already Casted or not?
<<Include>>
<<Extend>>
Store Vote
Cast Vote
<<Include>>
Entered Valid Candidate?
Login
<<Include>>
See vote Details
VOTER
Candidate Module
Election Commission Verification
<<Include>>
Registeration
already registered or not?
<<Include>>
Save account INFO
Create Login
<<Include>>
LOGIN
See votes secured
<<Include>>
Candidate
Admin
Admin Module
Manage NADR
LOGIN
Manage Election Commission
<<Include>>
Display Information
Add Information
Update Information
Delete Information
Verification
Display Information
Add Information
Update Information
Delete Information