

# LAB MANUAL

## List Of Experiments:

**Subject:** Object Oriented Software Engineering and Web Engineering

**Class:** T.Y.B.Tech(Computer) AY 2014-2015

EXP NO	CONTENTS
1	To select a case study and write a detailed problem statement.
2	To write a software requirements specification .
3	To draw Use Case Diagram for the case study only.
4	To draw Class Diagram for the case study only.
5	To draw State Transition Diagram for the case study only.
6	To draw Activity Diagram for the case study only.
7	To draw Interaction Diagram for the case study only.
8	To design Data Structure Design for the case study only.
9	To design Algorithm for the case study only.
10	To draw Component Diagram for the case study only.
11	To draw Deployment Diagram for the case study only.
12	To draw GUI for the case study only.

# ONLINE VOTING SYSTEM

## PROBLEM STATEMENT:

The Online voting system (OVS) also known as e-voting is a term encompassing several different types of voting embracing both electronic means of counting votes. Electronic voting technology can include punched cards, optical scan voting systems and specialized voting kiosks (including self contained direct- recording electronic voting systems or DRE). It can also involve transmission of ballots and votes via telephones, private computer networks, or the internet. Online voting is an electronic way of choosing leaders via a web driven application. The advantage of online voting over the common “queue method” is that the voters have the choice of voting at their own free time and there is reduced congestion. It also minimizes on errors of vote counting. The individual votes are submitted in a database which can be queried to find out who of the aspirants for a given post has the highest number of votes. This system is geared towards increasing the voting percentage in India since it has been noted that with the old voting method {the Queue System}, the voter turnout has been a wanting case. With system in place also, if high security is applied, cases of false votes shall be reduced.

Online voting system is an online voting technique. In this system, people who have citizenship of India and whose age is above 18 years of age and any sex can give his/her vote online without going to any physical polling station. There is a database which is maintained in which all the information about voters is stored.

In ONLINE VOTING SYSTEM, a voter can use his/her voting right online without any difficulty. He/she has to be registered first for voting purpose. Registration is mainly done by the system administrator for security reasons. The system administrator registers the voters on a special site of the system visited by him only by filling a simple registration form to register voter. Citizens seeking registration are expected to contact the system administrator to submit their details. After the validation of them being citizens of India has been confirmed by the system administrator by comparing their details submitted with those in existing databases such as those in Registrar of Persons, the citizen is then registered as a voter. After registration, the voter is assigned a secret Voter ID with which he/she can use to log into the system and enjoy services provided by the system such as voting. If invalid/wrong details are submitted, then the citizen is not registered to vote.

## **The objective presented as advantage is as follows:**

### **Saved Ballot Templates**

These eliminate the need to configure elections from scratch. Just do it once, then save that ballot configuration, and in subsequent years, specify only the names of the candidates.

### **Reduced costs**

Elections are enjoyed when the expenses of printing, mailing and tabulating paper ballots are lessened or even eliminated entirely from the election process.

### **Email Reminders**

It helps to increase voter participation by sending reminders to voters who've not voted and providing them with a link straight to the online ballot.

### **Ballot Shuffling**

This randomly orders the candidates' names on each request of the ballot so all candidates get a fair chance at prime name placement on the ballot.

### **On-demand Paper Ballots**

This simplify hybrid elections by allowing an election administrator to generate a voter-specific paper ballot that honors all the election settings just as web ballots do.

### **Automated Tallying**

Most importantly, it removes human fallibility from the tabulation process and makes your election results available within seconds of the close of the election.

### **Comprehensive Reporting**

It instantly provides you with informative day-by-day statistics about your election beyond just who won

### **Archived Election Results**

All previous election results are readily available at your account which saves you from having to dig through a file cabinet to find results from previous elections.

### **Ability to correct mistakes**

It allows voters to go back and correct any mistakes before final submission of their ballot. Once a ballot has been submitted however, it is final and can not be altered.

In short, an OTA attempts to provide a simplified user interface over very complex systems and data, and attempts to do so in an efficient and cost effective way that can scale to any number of users.

# **Software Requirements Specification**

**For**

## **Online Voting System**

**By**

**Rupali Gawali (131071041)**

**Chetana Patel(131071045)**

**Shama Kamat(131071056)**

**T.Y. B.Tech-Semester VI**

**VJTI, Mumbai**

## Table of Contents

<b>1.Introduction</b>	<b>5</b>
1.1Purpose	5
1.2Document Conventions	5
1.3Intended Audience and Reading Suggestions	5
1.4Product Scope	6
1.5References	6
<b>2.Overall Description</b>	<b>7</b>
2.1Product Perspective	7
2.2Product Features	8
2.3User Classes and Characteristics	10
2.4Operating Environment	10
2.5Design and Implementation Constraints	10
2.6User Documentation	10
2.7Assumptions and Dependencies	10
<b>3.System Features and Functional Specification</b>	<b>12</b>
<b>4.External Interface Requirements</b>	<b>15</b>
4.1User Interfaces	15
4.2Hardware Interfaces	21
4.3Software Interfaces	21
4.4Communications Interfaces	21
<b>5.Other Nonfunctional Requirements</b>	<b>22</b>
5.1Performance Requirements	22
5.2Safety Requirements	22
5.3Security Requirements	22
5.4Software Quality Attributes	22

## **1. Introduction**

This document describes the structural properties and software requirements of the Online National Election Voting System project.

### **1.1.Purpose**

Online Voting System is a system which enables all citizens to cast their vote online. The purpose is to increase the voting percentage across the country, as in the present system people have to visit the booth to cast their vote and those people who live out of their home town are not able to cast vote during the elections. So due to this the voting percentage across the country is very less. Through this software those people who live out of their home town will also be able to cast their votes as this system is online.

### **1.2.Document Conventions**

Throughout the document, the font used is Times New Roman and the font sizes are such that the topic is easily understood and followed as in the case of any conventional document practice. IEEE Standard Software Requirement Specification is used.

Title:

Font Style: Times New Roman

Size: 16

Heading:

Font Style: Times New Roman

Size: 16

Subheading:

Font Style: Times New Roman

Size: 14

Content:

Font Style: Times New Roman

Size: 12

### **1.3.Intended Audience and Reading Suggestions**

This document is intended for readers that include developers, project managers, users, students and researchers working in this domain. This SRS is categorized into five sections, viz. Introduction, Overall Description, System Features, External Interface Requirements, Other Non-functional Requirements and Other Requirements.

### **1.4.Product Scope**

The software produced will be an online voting system.The main objective of this software is to increase the overall voting percentage.It will maintain the database of all the eligible citizens and candidates.It will manage all the account details of the voters such as citizen name, date of birth, their constituency area, region, state, login id and password of the voter from one central location..

### **5. References**

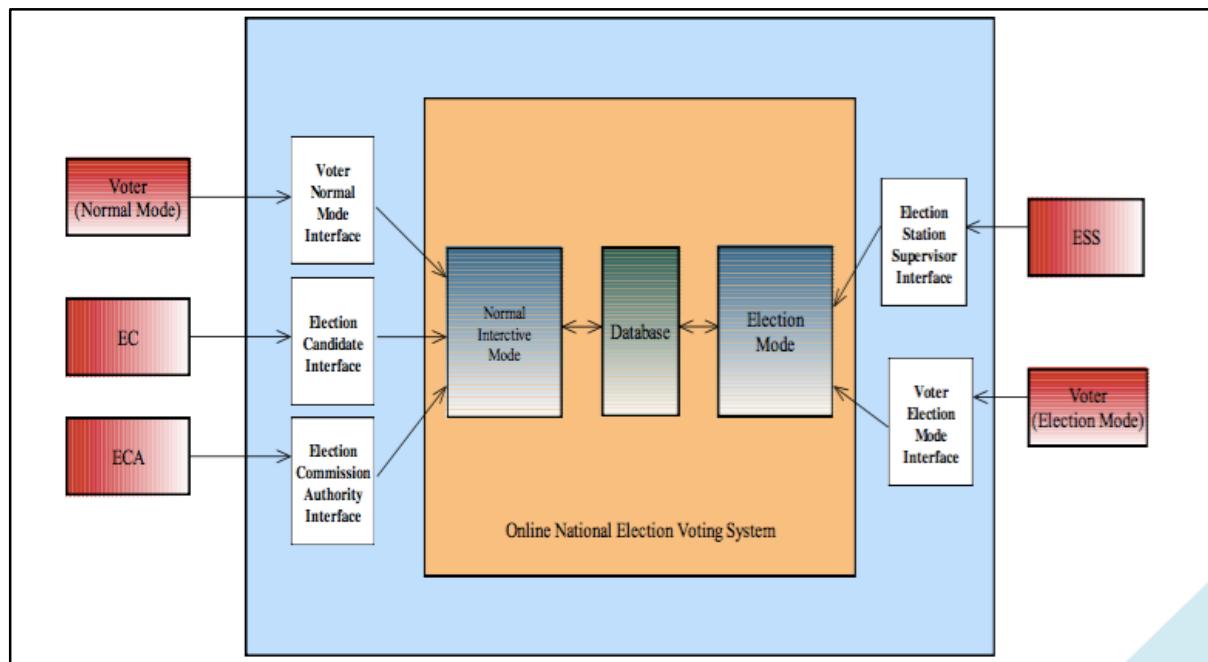
- 1) Standard IEEE 830 recommended practice for software requirements specifications
- 2) Software Engineering - Robert Pressman
- 3) Database System Concepts-Korth

## 2. Overall Description

### 2.1. Product Perspective

The software product is a standalone system and not apart of a larger system. The system will be made up of two parts, one running visible directly to the administrator on the server machine and the other visible to the end users, in this case the voters, through web pages. The two users of the system, namely the voters and the admin interact with the system in different ways. The admin configures the whole system according to its needs on the server where the system is running. The voters cast their votes using the web interface provided. These votes are accepted by the system on the server. The voters will reach the system through web pages by using web-browsers such as Mozilla, Internet Explorer and Google Chrome.

On the election day another independent system will be used for voting operations. This system will be adapted to the computers at the polling stations. The voters cast their votes using the interface that are provided at these machines. These votes are accepted by the system on the server. The ECA configures the whole system according to its needs on the server where the system is running.



**Block diagrams showing interaction between users and the systems**

## **2.2.Product Features**

The system can function in two modes, namely, Normal Interactive Mode and Election Mode. The system will be in Election Mode, for the purpose of vote polling only on the Election Day. Normal Interactive Mode is for accepting registrations, discussions between voters and candidates, campaigns and the system is available in this mode all the time except Election Days.

### **2.2.1. Normal Interactive Mode**

#### **2.2.1.1. Voter Registration**

That system will be used only by the people who have been registered to the system. Main actor of the registration operator is the voter. The registration operator is approved by the Election Commission Authority.

#### **2.2.1.2. Approve Applicant**

By using this function, Election Commission Authority approves the application sent by the voters in order to use the Online Voting System. The main actor is the Election Commission Authority.

#### **2.2.1.3. Update Registered Voters**

Election Commission Authority deletes voters from the system who cannot use their vote officially. Election Commission Authority also updates voter's information. The main actor is the Election Commission Authority.

#### **2.2.1.4. Open Candidate Account**

The Election Candidate profile must be created by the Election Commission Authority. This functionality helps to perform this action. The Election Commission Authority is the main actor of this functionality.

#### **2.2.1.5. Login/Logout**

All of the system users login to system by their user ids and passwords. All of the users are the main actor of this use case.

#### **2.2.1.6. Account Update**

By using this function the Election Candidate may change his password that enters the system. The main actor of this use case is the Election Candidate.

#### **2.2.1.7. View Election Candidate Information**

This function allows the voters to reach information about the Election Candidate such as their CVs, etc. Main actor is the voter for this use case.

### **2.2.1.8. CV Edit**

This function provides the Election Candidate to edit his CV information on his own profile. The EC is the main actor of that functionality.

### **2.2.1.9. View Election Results**

This functionality provides voters to see the current or past years' election result in a proper way. The main actor is the voter.

## **2.2.2. Election Mode**

### **2.2.2.1. Open System**

This function provides Election Station Supervisor to start the system during the Election Day or before. The Election Station Supervisor is the main actor of this operation.

### **2.2.2.2. Mark Generate Password**

By using that function the Election Station Supervisor will generate a password which will be used at voting operation by the voters. Main actor of this operation is the Election Station Supervisor.

### **2.2.2.3. Online Vote**

This is the main function of the system that provides online voting for the general public. The main actor is the voter and votes are collected in the Database.

### **2.2.2.4. Enter Offline Votes**

By using this function the Election Station Supervisor's enters the offline votes to the system. The main actor of this use case is the Election Station Supervisor's.

### **2.3. User Classes and Characteristics**

The users can be divided into two main classes:

#### **The Admin**

They manage the entire Voting System Software and Conduct the Elections. They act as the Election Authority.

#### **The Voters**

The voters should have a basic knowledge of how to use a web browser and navigate through web pages. The voters should be aware that they have to keep their user-id and password confidential

### **2.4. Operating Environment**

Hardware Requirements:

- Processor: Dual Core Intel Processor
- RAM: 512 MB or higher
- HDD: 1 GB or higher

Software Requirements:

- Technology: QT C++
- Operating system : Windows XP or Windows 7

### **2.5.Design and Implementation Constraints**

Only a registered voter can use the system. For registering the user needs to give the information accordingly asked by admin through the website. The new username and password will be generated.. Only the admin is allowed to modify the system database. He can add plans and services and display them on the website.

### **2.6.User Documentation**

The user will be provided with manual or user guide to understand how to use the system. These documents will be freely available on the website. And can also be downloaded on the mobile phone by requesting the admin.

### **2.7.Assumptions and Dependencies**

The system enables voters to poll their vote from any election centers that the system is installed in. Security and safety are the most crucial fundamentals of the Online Voting System. The

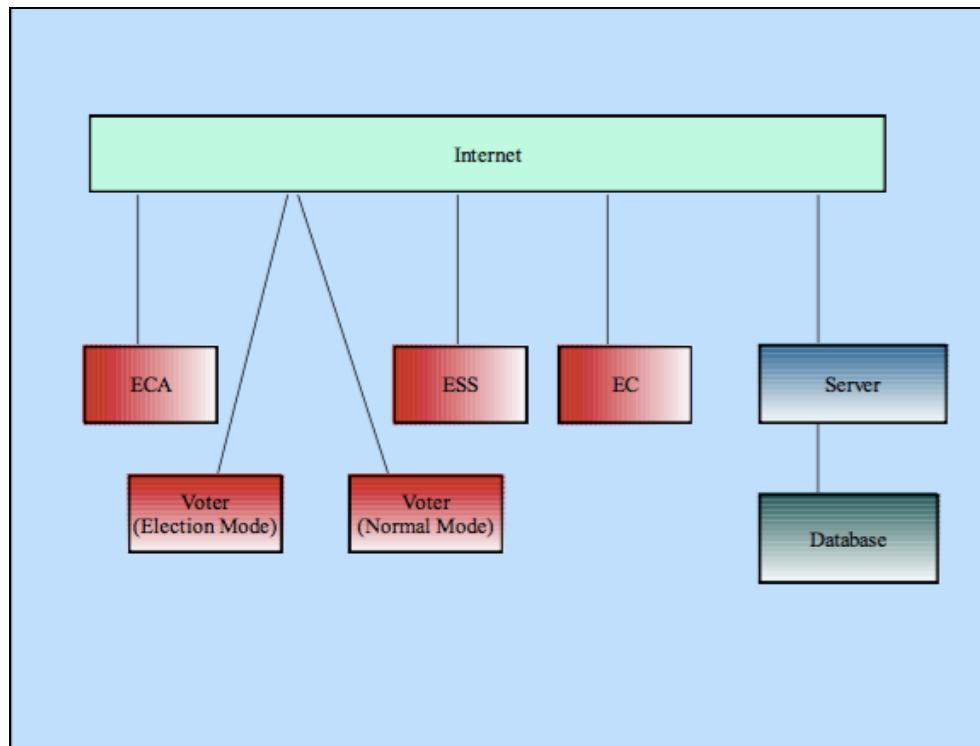
system has zero-tolerance with regard compromising. The system should not allow Election Station Supervisor to download votes to infer how voters in their regions have voted. The system should provide means for protecting and securing recounts of ballots cast in election.

For the proper working of the system we can list our assumptions and dependencies as follows.

- Working internet connection
- The voting results will be managed and calculated by system only.
- Administrator is created in the system already

### 3. System Features and Functional Specification

#### Interface requirements:



#### Functional Requirements:

##### 3.2.1. Normal Interactive Mode

This is a normal mode –before and after Election Day - a user interacts with the system. It involves registration for voting, updating profile, viewing election candidates (EC) . It also includes functions for the Election Commission Authority (ECA) to register EC and approve registered voters. The following use-cases describe the functional requirements.

###### 3.2.1.1. Voter Registration

Voter registers by filling the necessary information related to him/her in the application form and uploading a picture for Voter Identity Card.

###### 3.2.1.2. Approve Application

The application sent by voter is approved by Election Commission Authority and generates a new online account to this new voter and Voter Identity card is issued.

### **3.2.1.3. Update Registered Voters**

In order to update the voter information, Election Commission Authority checks the name in the list if the voter exists, the system updates the voter with respect to official the voter information.

### **3.2.1.4. Open Candidate Account**

ECA opens a new form for every official candidate and fills form according to candidate's information.

### **3.2.1.5. Log In /log out**

The user enters his login id and password

- A. If the login and password is valid, a session is opened
  - i. The security is verified
  - ii. The specific page of every user is loaded
- B. If the login or password is not valid, the login screen is redisplayed with an error message

The user click on the logout button

- i. The session is terminated.
- ii. The login screen is displayed.

### **3.2.1.6. Update Account**

1. In the user profile there is a button labeled "Update Account" he clicks it to update his account.

- The system opens a new page to enter old password and the new one.
- 2. The user enters his old password. He then enters his new password
- 3. User clicks the "submit" button.
  - If the old password was entered incorrectly, the system will print an error message and the form to change password will re-appear.
  - If the old password was entered correctly the system changes password and prints a success message and redirects to his profile.

### **3.2.1.7. View Election Candidate Information**

Voter selects the candidate information where candidate's profile is displayed along with the CV of candidate.

### **3.2.1.8. View Election Results**

The Voter can see the election results online by simply clicking the link

## **3.2.2. Election Mode**

### **3.2.2.1. Open System**

The user enters his/her VoterID and password

- A. If the VoterID and password is valid, a session is opened
  - I. The security is verified
  - II. The voting page is loaded
- B. If the VoterID and password is not valid, the login screen is redisplayed with an error message

### **3.2.2.2. Vote Online**

A voter enters his/her userid and password and proceed for voting online. The system will prompt a list of candidates. Voter chooses one of the candidates and votes for it.

## 4.External Interface Requirements

### 4.1 User Interfaces

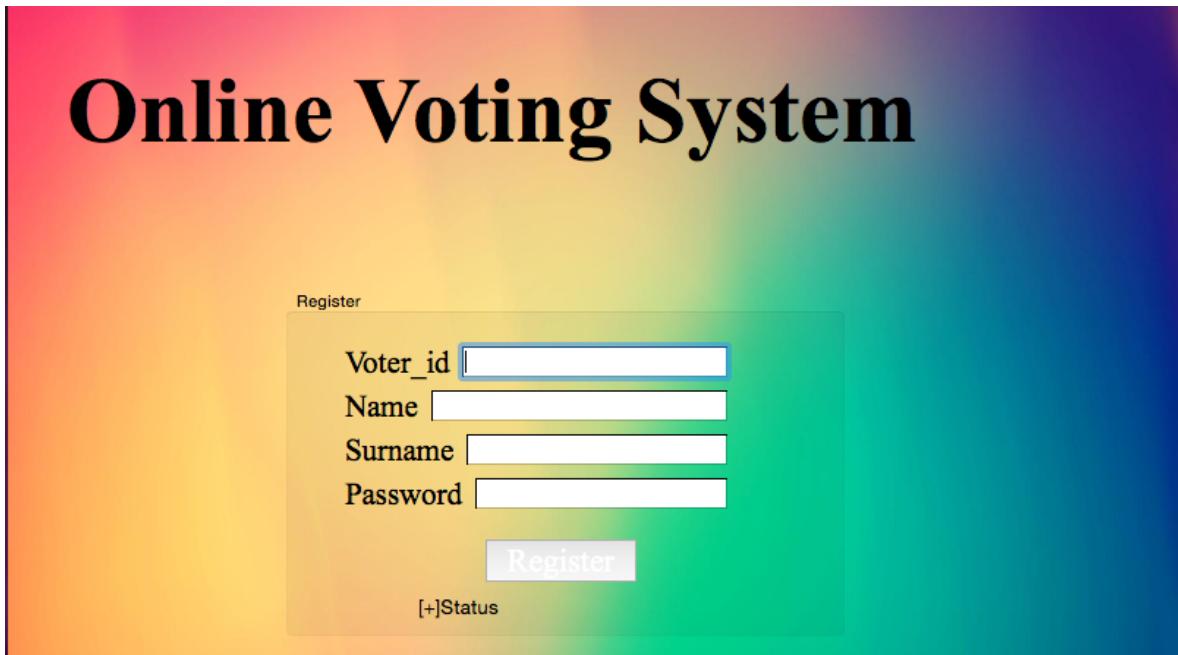
The website will provide the features of registration, logging in to user account or admin account, registration of the candidates and voters by admin, viewing candidates profile and voting by the voter. The voter can also report a complaint to the admin for the inconvenience faced.

1. The initial user interface is the registration of the voter with the admin, i.e.r registration by filling registration form and verifying required documents by admin.
2. Updating the candidate's information by admin.
3. Voters can view the candidate's profile and vote by logging in.
4. Lodging complaint

#### 1.Online Voting System MainWindow:



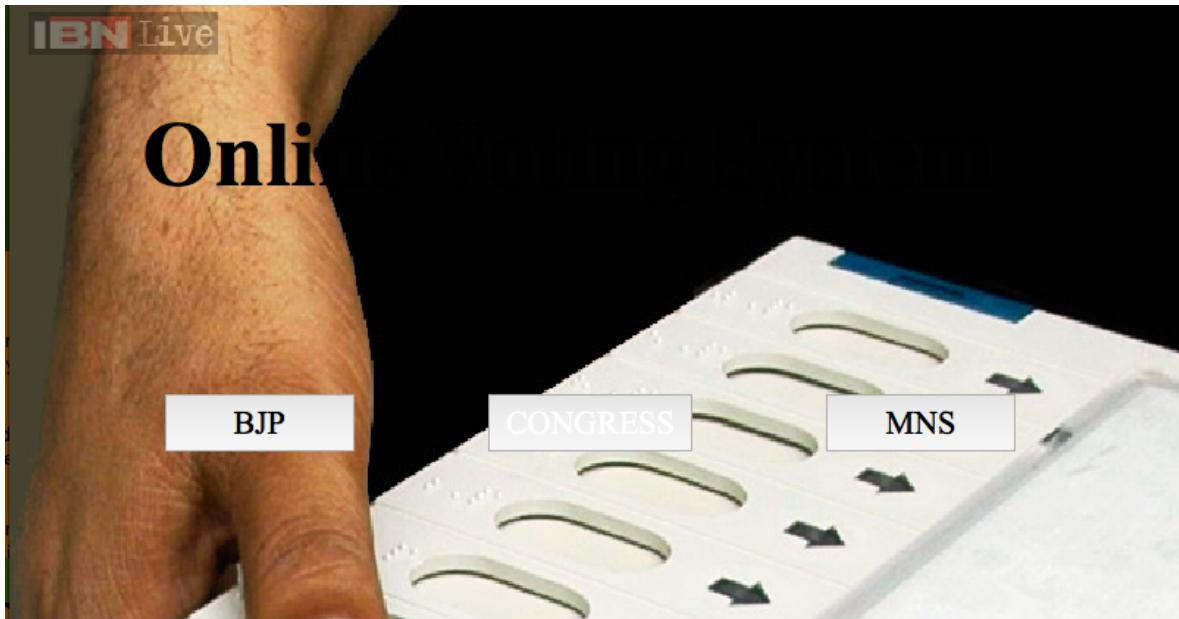
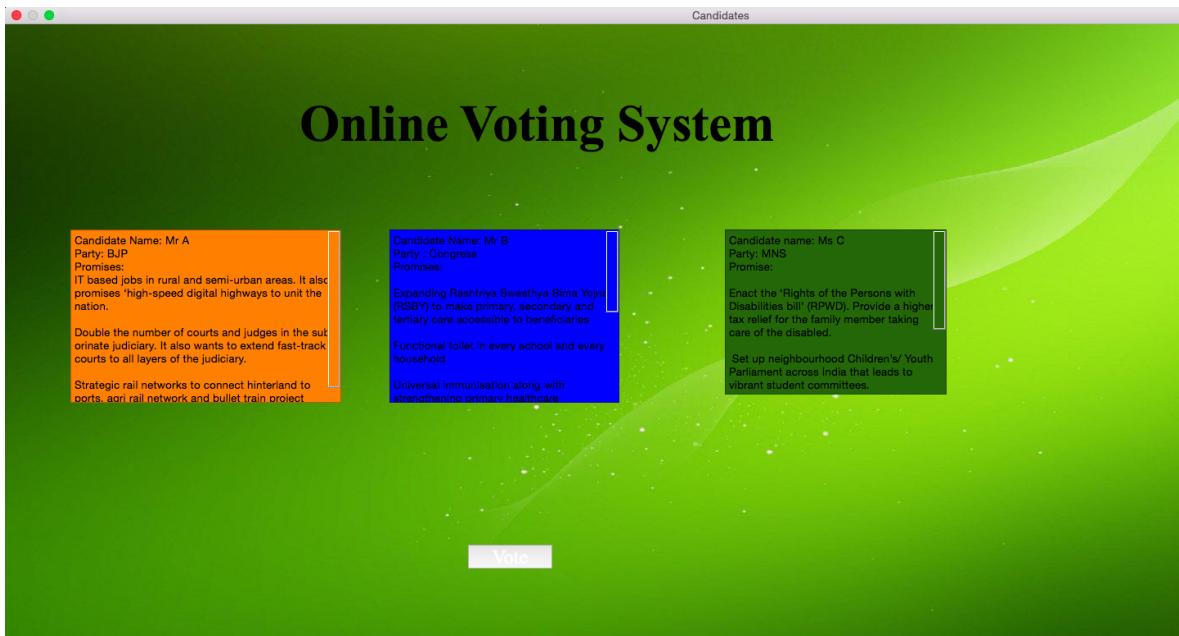
**1. Registration of voters:** Voter can register with admin by filling the registration form..



**2. Login :** A voter can login by Voter id and password given at the time of registration.



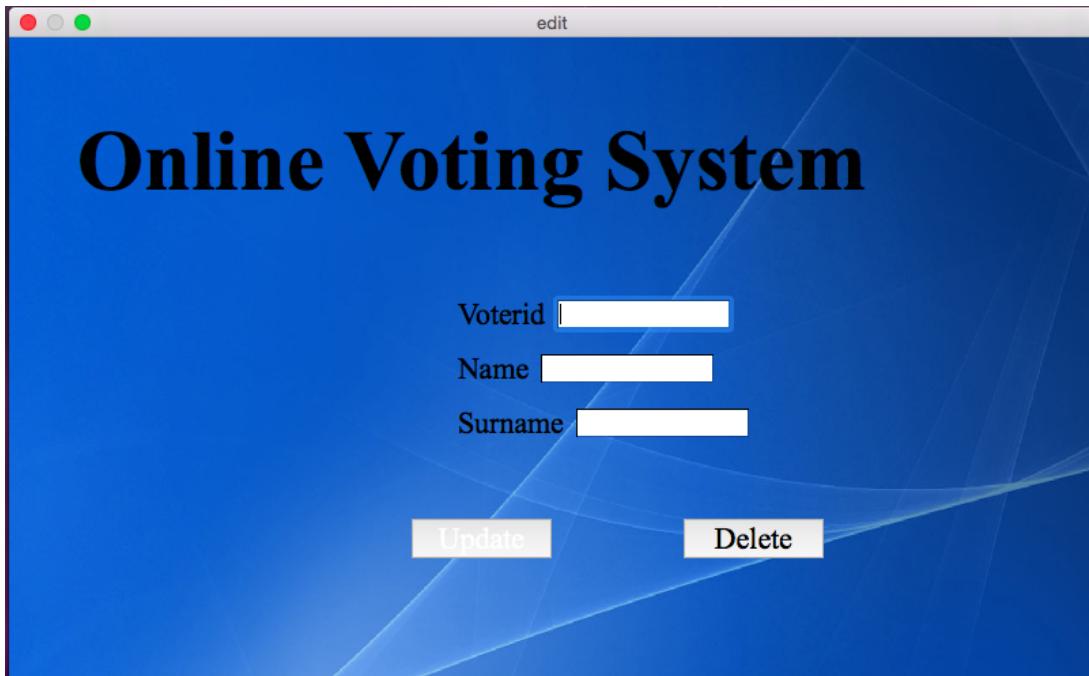
**3. View the candidate's information and vote:** Voters can see the candidate's information and vote the right person.



**4. Admin log in:** Admin can login by giving his username and password. So that he can update the candidate's information, voters database, etc.



**5.Edit Voter Information:** Editing can be done only by admin.



**6.Results:** Results of voting is displayed.



## Database Tables:

The screenshot shows the SQLite Database Browser interface with the database file 'OnlineVoting.sqlite' selected. The left sidebar lists the schema: Master Table (1), Tables (4), admin (selected), registerable, vote, voterlist, Views (0), Indexes (1), and Triggers (0). The right pane displays the 'admin' table structure with columns: TABLE, admin, Search, Show All, Add, Duplicate, Edit, Delete. The data table contains three rows:

rowid	username	password
1	Shama	shama
2	Rupali	rupali
3	Chetana	chetana

The screenshot shows the SQLite Database Browser interface with the database file 'OnlineVoting.sqlite' selected. The left sidebar lists the schema: Master Table (1), Tables (4), admin (selected), registerable (selected), voterid, name, surname, password, vote, candidatename, voterlist, Views (0), Indexes (1), and Triggers (0). The right pane displays the 'registerable' table structure with columns: TABLE, registerable, Search, Show All, Add, Duplicate, Edit, Delete. The data table contains eight rows:

voterid	name	surname	password
123	abc	bcd	efg
1310	chet	p	chet
131071026	Saiksha	Shetty	saiksha
131071031	Arsha	Khan	arsha
131071034	acv	asd	qwe
131071041	Rupali	Gawali	rupali
131071045	Chetana	Patel	chetana
131071056	Shama	Kamat	shama

The screenshot shows the SQLite Database Browser interface. On the left, the database structure is listed under 'OnlineVoting.sqlite'. It includes a 'Master Table (1)' and four tables: 'admin', 'registerable', 'voterid', and 'vote'. The 'vote' table is currently selected, indicated by an orange bar at the top of its list item. The main pane displays the 'vote' table with the following data:

TABLE	vote	Structure	Browse & Search	Execute SQL	DB Settings
rowid	candidatename	Add	Duplicate	Edit	Delete
1	CONGRESS				
2	CONGRESS				
3	BJP				
4	CONGRESS				
5	BJP				
6	CONGRESS				
7	BJP				

## 4.2 Hardware Interfaces

The hardware interface to be used for communication is the computer. A registered voter is given a username and password. Other hardware interfaces are the components of the internet connection architecture involved in actual functioning of the system.

## 4.3 Software Interfaces

Software requirement of the system are very nominal. The system does not require any additional except QT Creator and hence is economically feasible. Qt Creator is a cross-platform C++, JavaScript and QML integrated development environment which is part of the SDK for the Qt GUI Application development framework.

## 4.4 Communications Interfaces

The website is accessed using the web browser which uses the HTTP protocol. Voters need to browse the link through any web browser.

## 5.Other Nonfunctional Requirements

### 5.1Performance Requirements

The system should have a quick response time. In other words the confirmation of the voting should be received within 10 minutes of voting. The updates done to the database should quickly be reflected over the website. For e.g. when an admin updates the candidate's profile, it should be immediately reflected in the voters window highlighting the update in profile and the user should be able to view the updates. Web pages should be accessed immediately when the link is clicked. Delay should be tolerable.

### 5.2Safety Requirements

There should be a prior notice put on the website if the website is being updated and no transactions can be successful at that very time. If it so happens, there would be a problem during payment.

### 5.3Security Requirements

The user account and system database should be secured from tampering and should be accessible only to the admin. The security requirements for the system are as follows:

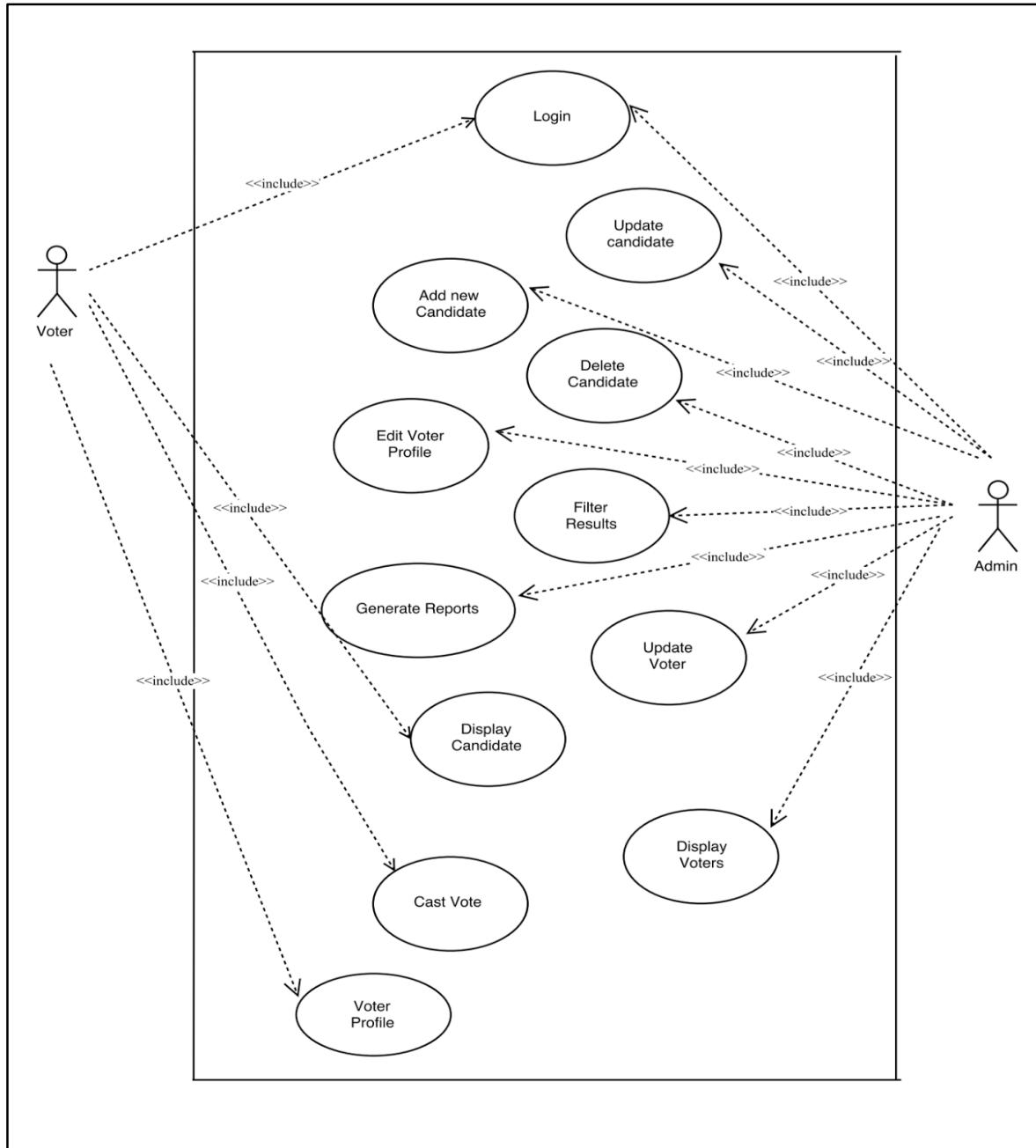
- **Access Control and authentication:** Only the voters who login and have a valid voter id and a password should be granted permission for voting.
- **Confidentiality:** All user data is required to be encrypted. Being transmitted, any attacker can hack this data and eavesdrop. Thus encryption is required to prevent any intrusion.
- **Anonymity:** Voting details of one voter should be hidden from another. Thus a voter is identified by his voter id and details.

### 5.4Software Quality Attributes

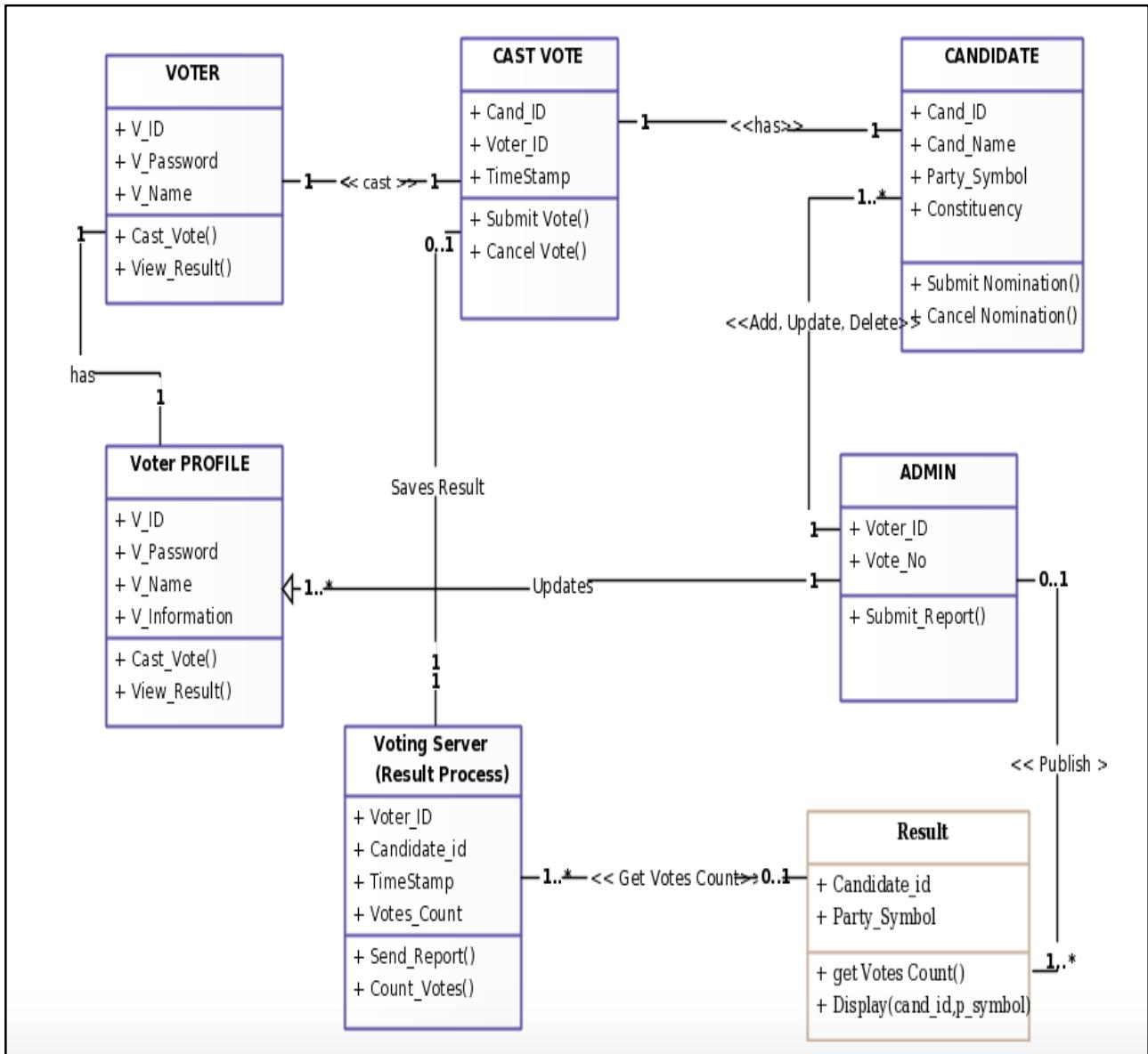
The quality of the system is maintained in such a manner that it would be user friendly and efficient. The following attributes are to be considered:

- **Availability:** The system should provide seamless connectivity to all internet users. Continuous updates should be minimized and website should be updated during midnight.
- **Portability:** Once you have an internet connection, you can easily access the site and get your voting done.
- **Usability:** The system should be user friendly, and easy to use.

### 3. Use case Diagram



## 4. Class Diagram



There are total 7 classes:

Voter, Cast Vote,Candidate ,Admin ,Result,Voting Server ,and Voter profile

## 5. State Transition Diagram

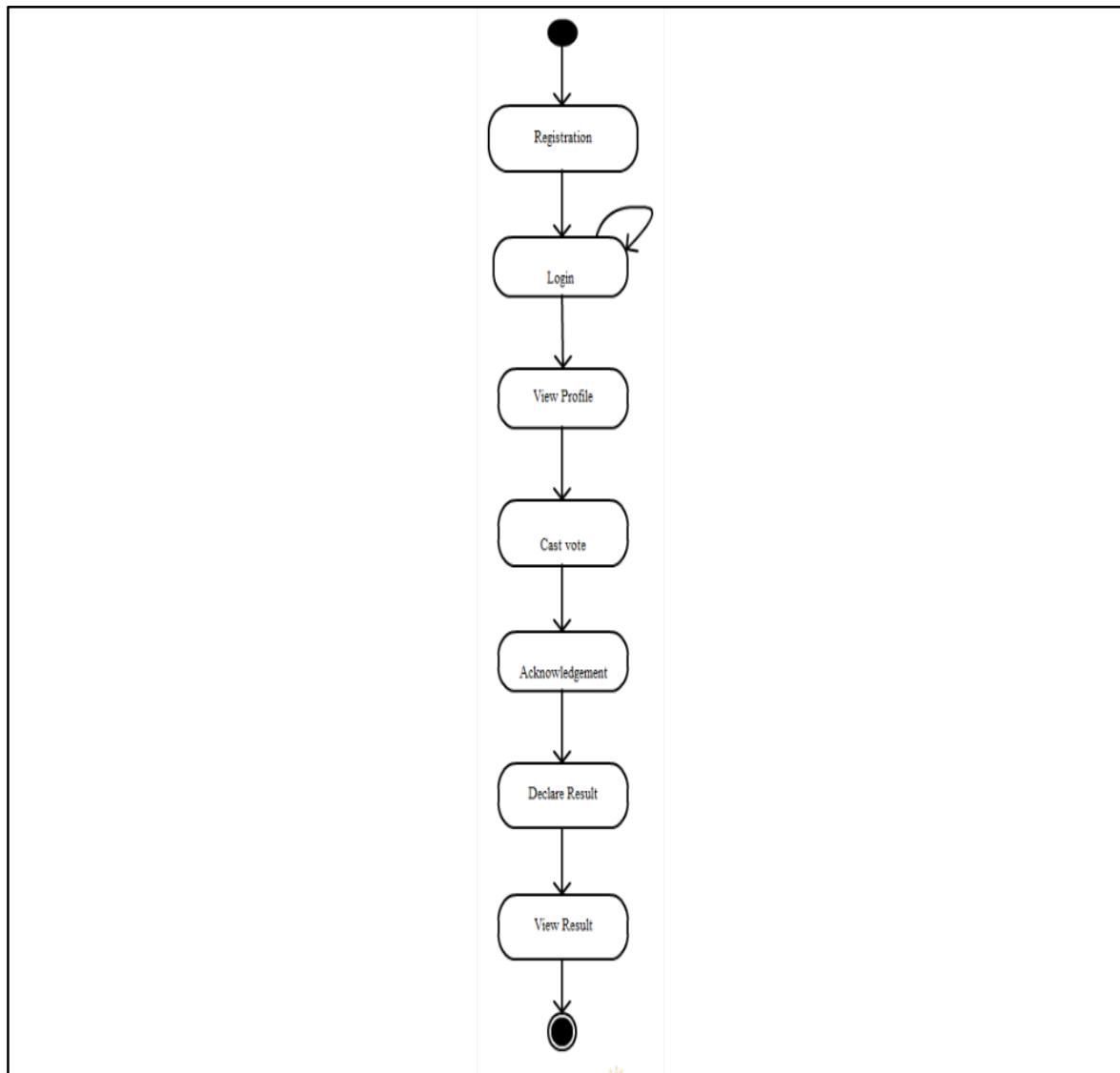
A state chart diagram shows the behaviour of classes in response to external stimuli. This diagram models the dynamic flow of control from state to state within a system. State diagrams require that the system described is composed of a finite number of states.

State diagrams are used to give an abstract description of the behavior of a system. This behaviour is analyzed and represented in series of events that could occur in one or more possible states. Hereby "each diagram usually represents objects of a single class and track the different states of its objects through the system".

States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State chart diagram is to model lifetime of an object from creation to termination. State chart diagrams are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.

Following are the main purposes of using State chart diagrams:

- To model dynamic aspect of a system.
- To model lifetime of a reactive system.
- To describe different states of an object during its lifetime.
- Define a state machine to model states of an object

**Description:**

1. In the shown State chart diagram, the system will initially remain in idle state.
2. Every user has to login before starting the functionality of the system hence at the time of login, system will remain in Authentication state.
3. After that user can caste vote, see candidate's details and see results.
4. The system then escorts the users on the website into the final state.

## 6. Activity Diagram

Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.

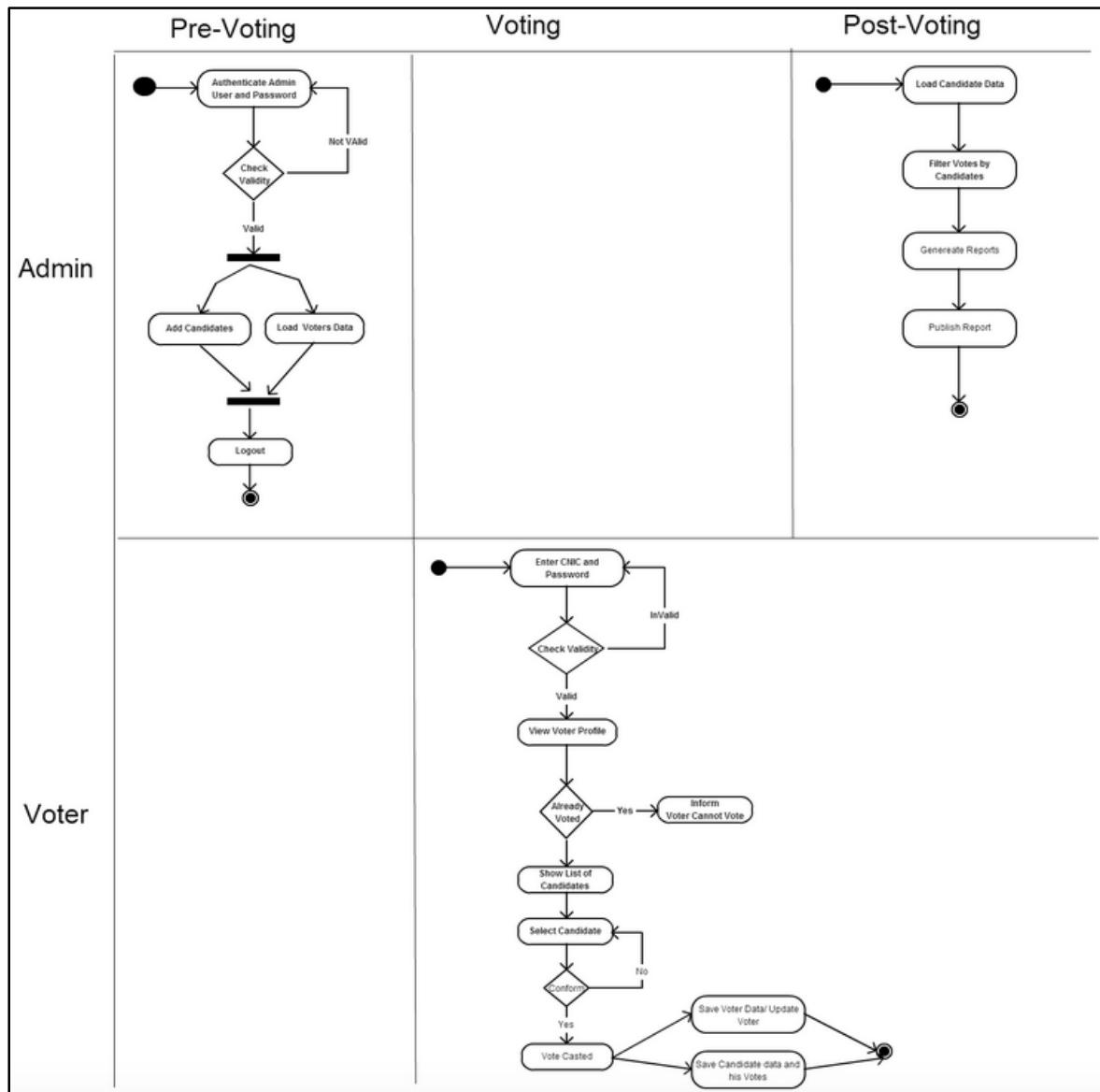
It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flow chart. Although the diagrams look like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

So the purposes can be described as:

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

### Description:

1. Activity Diagram for Online Voting System is shown in figure.
2. Here initially, user will login into the system.
3. As per login User the activities follows.
4. To cast the vote the voter has to go to voting page
5. Then the vote is given by voter
6. Voter can view candidate details then decide the vote
7. Then activities are terminated by LogOut procedure.



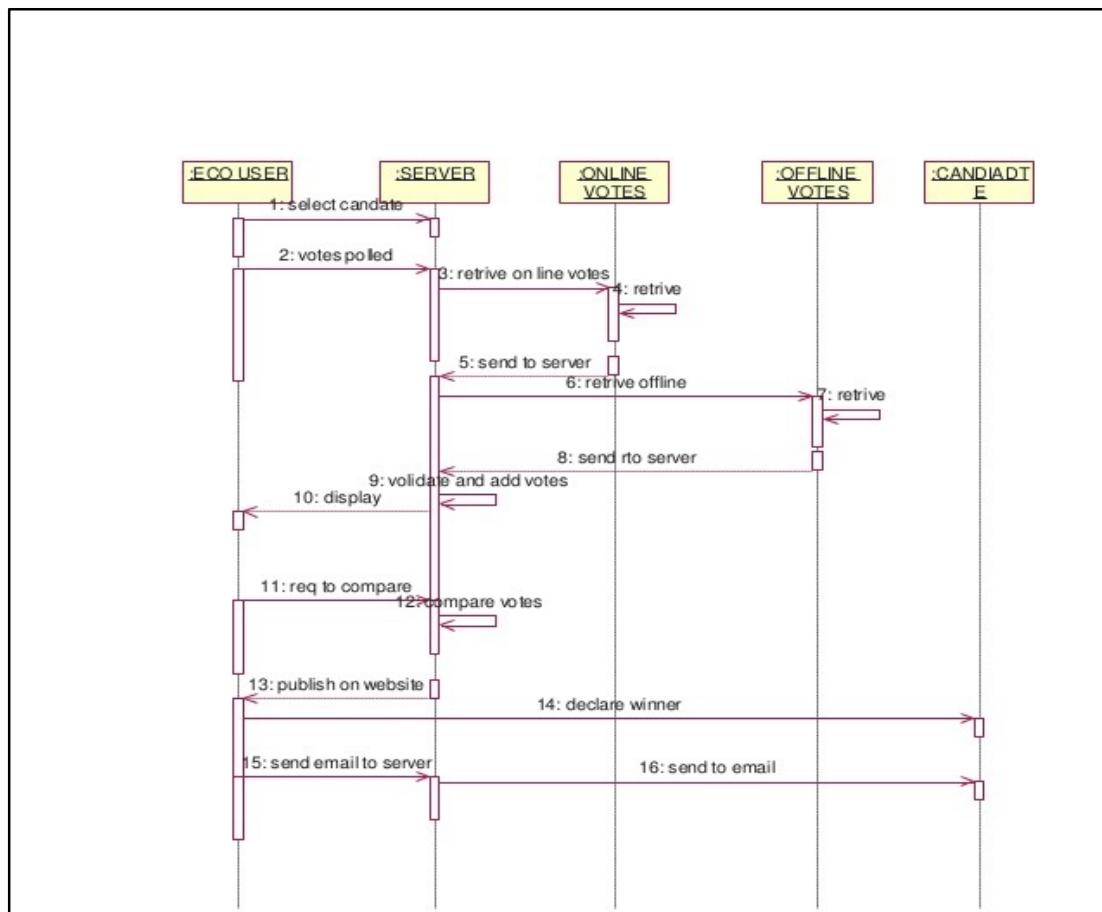
## 7. Interaction Diagram

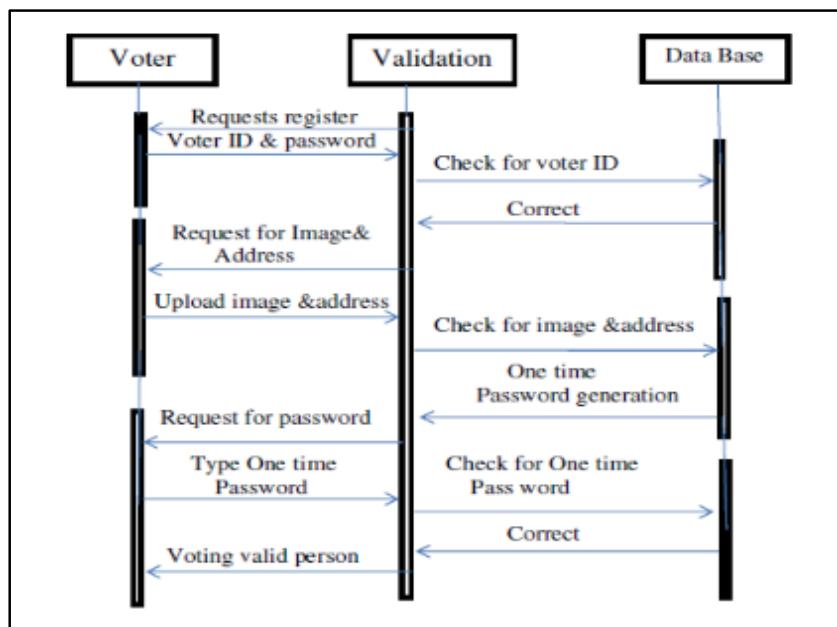
The purposes of interaction diagrams are to visualize the interactive behaviour of the system. Now visualizing interaction is a difficult task. So the solution is to use different types of models to capture the different aspects of the interaction.

That is why sequence and collaboration diagrams are used to capture dynamic nature but from a different angle.

So the purposes of interaction diagram can be described as:

- To capture dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe structural organization of the objects.
- To describe interaction among objects.





## 8. Data Structure Design

```
Class Voter{
    Private : String v_password
    Protected : int v_id;
    Public :String v_name;
        void cast_vote( );
        void view_result( );
    }
```

```
Class Candidate{
    Public : int cand_id;
        String cand_name;
        CImagepart_symbol;
        String constituency;
        void submit_nominations( );
        void cancel_nominations( );
    }
```

```
Class Cast_Vote{
    Protected : int cand_id;
        int voter_id;
    Public : Date timestamp;
        void submit_vote( );
        void cancel_vote( );
    }
```

```
Class Admin{
    Private : int voter_id;
    Protected : int voter_no;
    Public :void submit_report( );
    }
```

```
Class Voting_Server{  
Protected : int voter_jd;  
    int candidate_id;  
    int votes_count;  
Public :Date timestamp;  
    void send_report( );  
    void count_votes( );  
}
```

```
Class Result{  
Protected : int candidate_id;  
    CImage party_symbol;  
Public :void get_vote_counts( );  
    void display(cand_id, party_symbol );  
}
```

## 9. Algorithm Design

### 1. Login

**Description:** The user Login's into Online Voting System by providing appropriate Voter id and password.

**PreCondition:** A registered User.

**PostCondition:** The user is been authorized to further window if details are correct.

**Algorithm:**

1. Enter the username and password
2. Press login
3. If(username and password are valid)
4. Enter the user's account
5. else
6. Invalid username or Password
7. End if

### 2. Cast Vote

**Description:** The valid user casts his/her voter to any one candidate.

**PreCondition:** The user should be successfull with login

**PostCondition:** The user casts his vote.

**Algorithm:**

1. After successfull Login User clicks the voting tab.
2. The user clicks on candidate tab for which he/she wants to vote
3. Log out

### 3.Results

**Description:** The results of voting is displayed.

**PreCondition:** The results should be seen at the given date

**PostCondition:** The results are declared.

**Algorithm:**

1. Enter the username and password
2. Press login

3. If(username and password are valid)
4. Enter the user's account
5. Click on the result tab
6. Results Displayed
7. else
8. Invalid username or Password
9. End if

#### **4.Update Voter**

**Description:** The voter information is updated.

**Pre Condition:** The voter id should be registered

**PostCondition:** The voter details are updated.

**Algorithm:**

1. Admin enters the username and password
2. Press login
3. If(username and password are valid)
4. Admin Clicks Update button
5. Enter the Details of the voter whose account needs to be updated.
6. If(the voter Id is Registered)
7. Details Updated
8. Else
9. VoterID not registered.
10. else
11. Invalid admin name or Password
12. End if

#### **5.Delete Voter**

**Description:** The VoterID is deleted.

**Pre Condition:** The voter id should be registered

**PostCondition:** The voter details are deleted.

**Algorithm:**

1. Admin enters the username and password
2. Press login
3. If(username and password are valid)
  4. Admin Clicks Delete button
  5. Enter the VoterID whose account needs to be deleted.
  6. If(the voter Id is Registered)
    7. Account Deleted
    8. Else
    9. VoterID not registered.
  10. else
  11. Invalid admin name or Password
  12. End if

## 10. Component Design

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

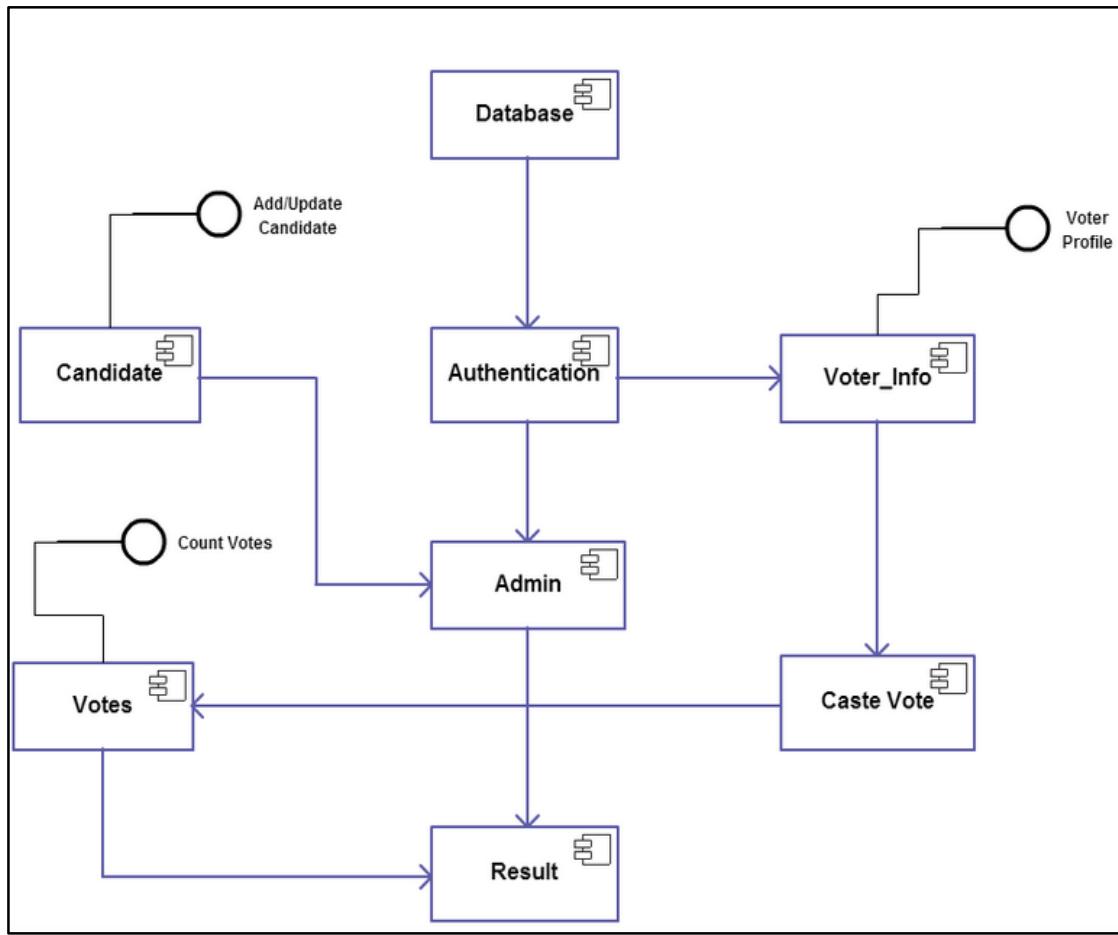
A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

So the purpose of the component diagram can be summarized as:

- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.

### Description:

The system consists of important components- Database, Authentication, Voter info, caste vote, Admin, Votes, Candidate, Results. The User will login through the Login component and can view and vote Candidate through cast vote and candidate information component. He can see the results once the voting is completed. All these components are connected with Database component via System package.



## 11. Deployment Design

The name *Deployment* itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components.

So most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.

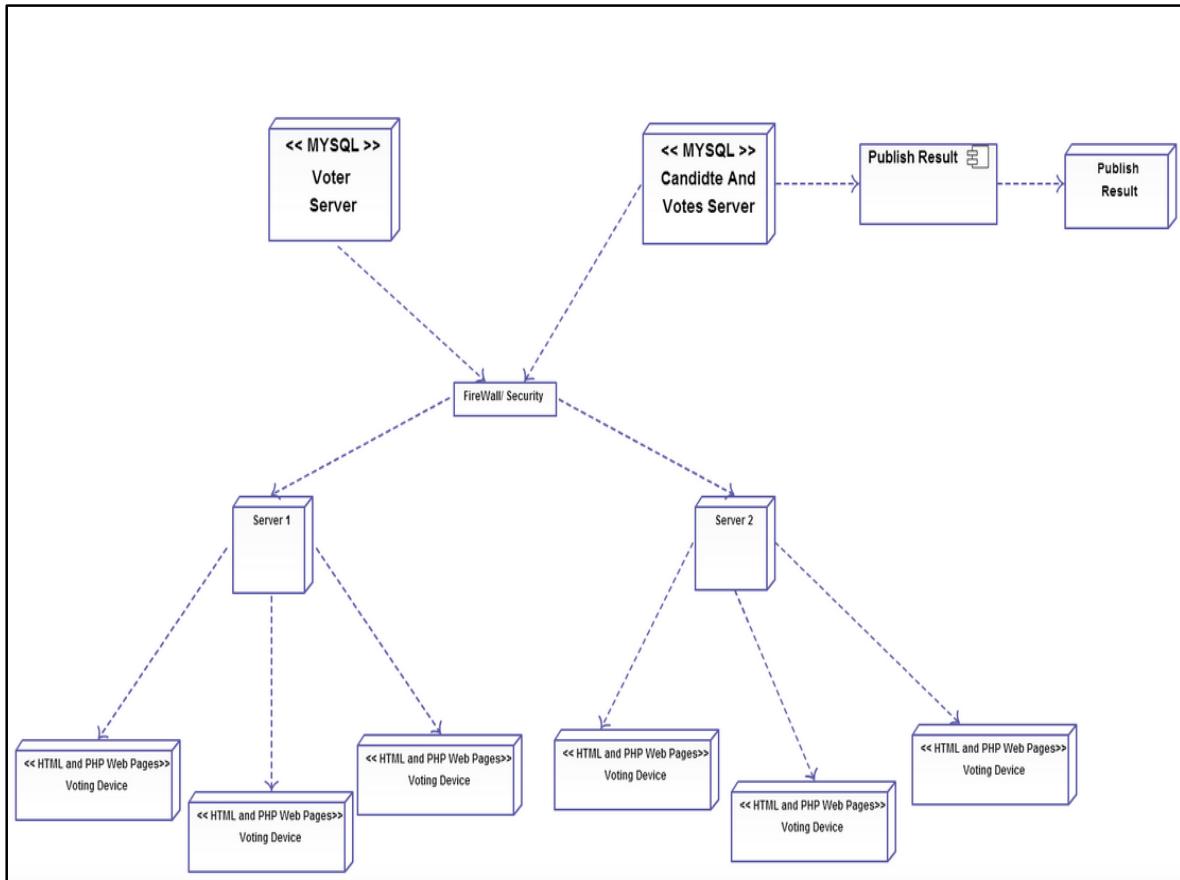
The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes.

### Description:

Fig. shows deployment diagram for Online Voting System. Various users uses the system through web. User will use their browsers for accessing the system. User will be connected to the system via internet connection which uses computer, modem and web access. The request from user will come to Web Servers. Web servers will process the request and respond to user via HTTP connection. Web Servers will user Database servers also which is developed by using MYSQL.

Web Servers will consist of Presentation Logic, Database Interface and Source Files. Here system requires Net framework and MYSQL as database system.



## 12. GUI

